



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**ANALÝZA VLASTNOSTÍ DASHBOARDŮ ZALOŽENÁ
NA AUTOMATICKÉ DEKOMPOZICI OBRAZOVKY**

ANALYSIS OF DASHBOARD ATTRIBUTES BASED ON AUTOMATIC DECOMPOSITION OF SCREEN

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. SANTIAGO MEJÍA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ HYNEK

BRNO 2017

Zadání diplomové práce

Řešitel: **Mejía Santiago, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Analýza vlastností nástroje dashboard založená na automatické dekompozici obrazovky**

Analysis of Dashboard Attributes Based on Automatic Decomposition of Screen

Kategorie: Uživatelská rozhraní

Pokyny:

1. Seznamte se vizualizačním nástrojem *dashboard*.
2. Dle pokynů vedoucího prostudujte vybrané metriky pro měření vlastností uživatelských rozhraní založených na hodnocení grafických objektů obsažených na obrazovce.
3. Dle pokynů vedoucího se seznamte s vybranými metodami pro automatickou dekompozici obrazovky uživatelských rozhraní.
4. Navrhněte způsob vyhodnocování vlastností nástroje dashboard využívající zvolené metriky z bodu 2 v kombinaci se zvolenou metodou automatické dekompozice obrazovky z bodu 3.
5. Implementujte navržený způsob vyhodnocování.
6. Otestujte implementovaný algoritmus na dostatečném množství vstupů a porovnejte s hodnocením reálných uživatelů. Zhodnoťte dosažené výsledky.

Literatura:

- Few, S.: *Information Dashboard Design: The Effective Visual Communication of Data*. Sebastopol [MA]: O'Reilly, 2006, ISBN 978-059-6100-162.
- Reinecke, K., et al: *Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hynek Jiří, Ing., UIFS FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 23. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Cieľom tejto práce je navrhnúť metódu automatickej segmentácie dashboardov tak, aby ich bolo možné analyzovať pomocou metrík. Tieto metriky analyzujú vlastnosti objektov obrazovky podľa regiónov. Táto metóda je založená na prístupu segmentácie zdola nahor a vytvára objekty na základe blízkosti. Hlavným výsledkom je rýchlejšia a presnejšia analýza dashboardov, nakoľko nie je potrebné ručne segmentovať obrázok.

Abstract

The aim of this paper is to propose a method for automatic segmentation of dashboards so that they can be analyzed using aesthetic by means of metrics. These metrics analyze the properties of the screen objects by region. This method is based on bottom-up segmentation method and creates objects based on proximity. The main result is a faster and more accurate analysis of dashboards, as there is no need to manually segment the image.

Kľúčové slová

Dashboard, užívateľské rozhrania, segmentácia dokumentov, metriky rozvrhnutia

Keywords

Dashboard, user interface, document segmentation, layout metrics

Citácia

MEJÍA, Santiago. *Analýza vlastností dashboardů založená na automatické dekompozici obrazovky*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek

Analýza vlastností dashboardů založená na automatické dekompozici obrazovky

Prehlásenie

Prehlasujem, že som tento semestrálny projekt vypracoval samostatne pod vedením pána Ing. Jířího Hyneka. Uviedol som všetky literárne pramene a publikácie, zo ktorých som čerpal.

.....

Santiago Mejía

23. mája 2018

Podakovanie

Touto cestou by som sa chcel poďakovať vedúcemu tohto semestrálneho projektu Ing. Jířímu Hynekovi zato, že vždy bol ochotný mi poradiť keď som sa dostal do ťažkostí a vedel ma podporiť v mojej práci.

Obsah

1	Úvod	3
2	Dashboard	5
2.1	Definícia nástroju dashboard	5
2.2	Kategórie dashboardov	7
2.2.1	Klasifikácia dashboardov podľa roly	7
2.3	Efektívne médiá pre zobrazenie dát na dashboardoch	8
2.3.1	Zbierka ideálnych médií pre zobrazenie dát na dashboardoch	8
2.4	Časté problémy v návrhu dashboardov	9
3	Metriky pre meranie vlastností užívateľských rozhraní	12
3.1	Popis metrík	12
3.2	Využitie metrík v súčasnosti	17
4	Vizuálne vnímanie objektov v užívateľských rozhraniach	18
4.1	Obmedzenia krátkodobej pamäte	18
4.2	Atribúty podvedomého rozpoznávania objektov	19
4.3	Gestalt princípy vizuálneho vnímania	21
4.3.1	Princíp blízkosti	21
4.3.2	Princíp podobnosti	22
4.3.3	Princíp uzavretia (enclosure)	22
4.3.4	Princíp uzavretia (closure)	23
4.3.5	Princíp kontinuity	23
4.3.6	Princíp pripojenia	24
4.4	Výskum rozpoznávania objektov	24
5	Metódy pre automatickú segmentáciu dokumentov	26
5.1	Metóda X-Y cut	27
5.1.1	Vylepšenie s použitím ohraničujúcich rámcikov	30
5.1.2	Vylepšenie segmentácie podľa obrysu využitím X-Y cut	30
5.2	Zdola nahor (Metóda RLSA)	31
6	Návrh metódy pre automatickú dekompozíciu dashboardov	33
6.1	Návrh metódy segmentácie dashboardov	33
6.2	Nástroj na vyhodnotenie dashboardov	35
6.3	Formát XML výsledku segmentácie	36
6.4	Návrh testovania a overenia výsledkov	37

7 Implementácia	39
7.1 Použité nástroje	39
7.2 Metódy automatickej dekompozície a predspracovanie	39
7.2.1 Predspracovanie	39
7.2.2 Hlavná funkcia	40
7.2.3 Utility použité pri segmentácii	43
8 Testovanie a vyhodnotenie	44
8.1 Testovanie	44
8.2 Vyhodnotenie výsledkov	44
8.2.1 Vyhodnotenie porovnaním regiónov	45
8.2.2 Vyhodnotenie porovnaním výsledkov metrík	47
9 Záver	49
Literatúra	50
Prílohy	52
Zoznam príloh	53
A Obsah priloženého pamäťového médiá	54
B Ukážka XML popisu výsledku segmentácie	55
C Ukážky automatickej segmentácie	56

Kapitola 1

Úvod

Vizualizačný nástroj *Dashboard*¹ zobrazuje najdôležitejšie informácie, ktoré sú potrebné k splneniu nejakého cieľa (využitie napr. v business intelligence [4]), pričom dôraz sa kladie na zrozumiteľnosť, jednoduchosť a prehľadnosť. Údaje sú zobrazené na jednej obrazovke, väčšinou v grafickej podobe.

Dashboard môže byť veľmi silným, efektívnym komunikačným prostriedkom, pokiaľ je dobre navrhnutý. Navrhnuť dashboard nie je vždy jednoduché. Je náročné splniť všetky požiadavky správne navrhnutého dashboardu. Návrhári dashboardov sa často dopúšťajú rôznych chýb, ktoré komplikujú vnímanie obrazovky užívateľom. Preto vznikla potreba vyhodnocovať návrhy dashboardov. Existuje niekoľko možností na vyhodnotenie návrhu dashboardov.

- Ohodnotenie cieľovými užívateľmi – spoľahlivý, ale časovo a finančne náročný spôsob, je potrebné mať hotový prototyp, ktorý užívateľ môže vyskúšať.
- Ohodnotenie na základe metrík – menej kvalitný prístup, je zameraný na základné vlastnosti ako usporiadanie, farby, atď. Oproti vyhodnoteniu cieľovým užívateľom je ale lacnejší a automatizovateľný.

Existujú metriky, ktoré vyhodnocujú na základe objektov/grafických elementov (*widgets*) kvalitu užívateľského rozhrania, tzv. *object-based metrics*. Tieto metriky vyžadujú ako vstup obálky grafických elementov obrazovky (regióny). Existujú rôzne možnosti, ako tieto regióny získať:

- Získať regióny zo zdrojových kódov webových stránok – nemusí odpovedať ľudskému vnímaniu obrazu. Kód nemusí byť dostupný. Existuje rôzne množstvo popisov, preto je nevyhnutné mať parser pre daný popis.
- Získať regióny od užívateľa – časovo náročnejší, subjektívny (získané objekty odpovedajú konkrétnemu užívateľovi).
- Detekcia regiónov v obrázku (bitmap) – jednotný popis (napr. bitmapa - RGB), výsledná podoba zobrazená na obrazovke, je nutné rozpoznávať v súlade s princípmi vnímania.

Táto práca sa zaoberá získavaním regiónov z bitmap. Cieľom je detekovať v dashboardu regióny reprezentujúce obálky vizuálne dominantných objektov. Výstupom je sada regiónov.

¹ Poznámka: Slovo *Dashboard* nie je slovenské slovo, ale ohľadom k častému použitiu ho zavediem a budem ho skloňovať podľa vzoru *dub*.

Tieto regióny slúžia ako vstup pre metriky na vyhodnotenie dashboardov. Dôraz sa kládol na to, aby výsledný popis regiónov odpovedal tomu, ako ich vidia ľudia. Bolo ukázané, že každý človek ich môže rozpoznávať trochu iným spôsobom (subjektívne vnímanie). Algoritmus bol vytvorený s ohľadom na Gestalt princípy a existujúcu štatistiku rozpoznávania regiónov užívateľmi vo vybranej vzorke dashboardov. Algoritmus vychádza z metódy segmentácie z dola nahor. Na základe blízkosti a podobnosti spája objekty do riadkov a stĺpcov a postupne vytvára väčšie celky. Podľa veľkosti sú tieto regióny rozdelené na dve skupiny. Menšie regióny sa ďalej skúmajú a podľa možnosti sú ďalej pospájané.

S dashboardami sa podrobnejšie zaoberá kapitola 2. Okrem samotnej definície a popisu dashboardu obsahuje aj kategorizáciu, dôležité vlastnosti a časté chyby dashboardov. Využitie celkovej plochy obrazu sa môže charakterizovať na základe rozličných metrík. Popisu metrík sa venuje kapitola 3, v ktorej sú tiež uvedené ukážky dobrých a zlých príkladov rozloženia užívateľských rozhraní podľa daných metrík. Kapitola 4 popisuje vizuálne vnímanie, ktoré ovplyvňuje návrh dashboardov ako limitácie ľudského mozgu, vizuálne rýchlejšie vnímateľné vzory a Gestalt princípy. Jedným z dôležitých krokov analýzy dashboardov je dekompozícia obrázkov na obdĺžniky, ktoré sa dajú vyhodnotiť podľa metrík z kapitoly 3. S týmito dekompozičnými metódami sa zaoberá kapitola 5, ktorá ďalej obsahuje aj možné vylepšenia a využitie v iných kombinovaných metódach. Kapitola 6 uvádza popis nástroju vyhodnotenia dashboardov, návrh algoritmu segmentácie dashboardov využitím metód z kapitoly 5 a návrh testovania metódy a spôsob vyhodnotenia výsledkov. Kapitola 7 popisuje implementáciu metódy automatickej dekompozície. Obsahuje popis hlavnej funkcie metódy a popis použitých utilít na spracovanie matíc. Kapitola 8 uvádza opis spôsobu testovania a vyhodnotenie dosiahnutých výsledkov.

Kapitola 2

Dashboard

V tejto kapitole si objasníme čo je dashboard. Uvedieme definíciu, efektívne médiá pre zobrazenie dát, kategorizáciu a najčastejšie problémy pri návrhu dashboardov.

2.1 Definícia nástroju dashboard

Množstvo informácií, ktoré nás obklopujú je ťažké efektívne spracovať a zobraziť, v dôsledku čoho vzniklo niekoľko spôsobov zobrazenia informácií. Jedným z týchto spôsobov je dashboard. Na obrázku 2.1 je vidieť príklad dashboardu.



Obr. 2.1: Príklad dashboardu, ktorý zobrazuje štatistiku návštevnosti webovej stránky.¹

¹Zdroj: <https://www.datapine.com/dashboard-examples-and-templates/marketing>

Ako môžeme vidieť z príkladov dashboardov v tejto kapitole, dashboardy sa môžu líšiť z rôznych pohľadov. Predajcovia softvéru pre tvorbu dashboardov majú odlišné predstavy o tom, čo všetko patrí na dashboard a definuje ich. Jedna z mála vlastností dashboardov, s ktorým väčšina predajcov súhlasí je, že k tomu, aby niečo mohlo byť nazvané dashboardom, musí obsahovať grafické zobrazovacie mechanizmy (napr. obrázok 2.4). Ďalšia vlastnosť, ktorú si môžeme všimnúť z obrázkov je, že všetky dashboardy sa snažia zobrazit prehľad niečoho, čo sa práve deje v danom priemysle. Z toho vyplýva, že neexistuje jednoznačná definícia dashboardov.

Jedna z možných definícií dashboardu je uvedená v knihe *Information Dashboard Design* [5]. Autorom knihy je Stephen Few a definícia môže byť preložená nasledovne:

„Dashboard je vizuálne zobrazenie najdôležitejších informácií potrebných na dosiahnutie jedného alebo viacerých cieľov, konsolidované a usporiadané na jednej obrazovke tak, aby bolo možné informácie jednoducho sledovať.”

Všeobecne platí, že dashboard je typ zobrazenia alebo forma prezentácie pre komunikáciu informácií a nie špecifický typ informácie alebo technológie. Charakteristické body dashboardov sú:

- **Dashboardy zobrazujú informácie potrebné k dosiahnutiu konkrétnych cieľov.**

K dosiahnutiu, niekedy aj jednoduchých cieľov potrebujeme zbierku informácií, ktoré spolu nemusia súvisieť alebo môžu pochádzať aj z rôznych zdrojov. Tieto informácie nemusia byť jedného špecifického typu, ale môžu byť informácie rôznych typov potrebných k danej úlohe. Informácie nemusia byť určené len pre vedúcich alebo manažérov ale aj pre ľudí, ktorý pracujú na danej úlohe.

- **Dashboard by sa mal zmestiť na jednu obrazovku počítača.**

Informácie by sa mali zmestiť na jednu obrazovku tak, aby boli úplne k dispozícii v rozpätí oka diváka a aby všetko bolo možné vidieť naraz na prvý pohľad. Ak je potrebné rolovať obrazovku k prehliadnutiu všetkých informácií, tak sa prekročili hranice dashboardu. Ak je potrebné prejsť z jednej obrazovky na druhú obrazovku, tak sa jedná o viaceré dashboardy. Cieľom je mať najdôležitejšie informácie k dispozícii, dostupné ľahko a bez námahy, tak aby sa mohli rýchlo absorbovať.

- **Je potrebné zobrazit dashboard vo webovom prehliadači?**

Webový prehliadač síce nie je jediným prostriedkom zobrazenia dashboardu, ale dnes je asi najvhodnejším. Veľký vplyv na to má faktor, ako často je potrebné informácie na dashboarde aktualizovať. Ak sa jedná o sledovanie informácií v reálnom čase, je očakávané prostredie s vyhovujúcimi vlastnosťami.

- **Dashboardy sa používajú na sledovanie informácií na prvý pohľad.**

Napriek tomu, že takmer hocijaké informácie môžu byť vhodne zobrazené na dashboarde, je potrebné sa zamerať iba na tie najdôležitejšie. Nakoľko dashboardy nemôžu obsahovať všetky detaily potrebné k dosiahnutiu cieľa, tieto informácie treba skrátiť formou zhrnutia alebo výnimky. Dashboard musí byť schopný zdôrazniť časti, ktoré vyžadujú pozornosť, prípadne ďalšie akcie. V takomto prípade sa nemusia zobrazovať všetky informácie potrebné k vykonaniu akcií, ale môže umožniť rýchlo a jednoducho sa dostať k týmto informáciám, napríklad odkazom na inú stránku. Toto môže znížiť ako porušenie základných pravidiel dashboardov, uvedených vyššie. Napriek tomu

takýto dashboard spĺňa svoju primárnu úlohu, čo je ukázať na to, kde je potrebné vykonať akciu.

Vyššie uvedené body sú podstatou dashboardov. Ďalej uvedieme vlastnosti, dodržanie ktorých by malo podporovať efektivitu dashboardov.

- **Dashboards majú malé, stručné, jasné a intuitívne zobrazovacie mechanizmy.**

Zobrazujúce mechanizmy by mali jasne uvádzať svoj obsah bez toho, aby zaberali veľa miesta. Je potrebné, aby celá zbierka informácií sa zmestila do obmedzenia jednej obrazovky. Mali by obsahovať prevažne grafickú prezentáciu zobrazenia, ktorá dokáže pojať väčšie množstvo informácií na malom priestore a dokáže zdôrazniť vzťahy medzi hodnotami lepšie ako textová prezentácia.

- **Dashboards sú prispôsobené.**

Informácie na dashboardoch musia byť prispôsobené požiadavkám daného človeka, skupiny alebo funkcie, inak nebude slúžiť svojmu účelu.

2.2 Kategórie dashboardov

Dashboards môžu byť kategorizované do skupín niekoľkými spôsobmi. Je to užitočné hlavne preto, lebo takto sa ukážu niektoré výhody a dôležité vlastnosti daných typov dashboardov. Tabuľka 2.1 uvádza kategorizáciu dashboardov podľa autora Stephen Few.

Premenné	Hodnoty
Role	Strategický, Analytický, Operatívny
Podľa typu dát	Spočetná, Nespočetná
Dátová doména	Obchodný, Financie, Marketing, Výrobné, Ľudské zdroje
Rozsah údajov	Celopodnikové, Odborový, Individuálny
Frekvencia aktualizácie	Mesačné, Týždenné, Denné, Hodinové, ...
Interaktivita	Statické zobrazenie, Interaktívny displej (napr. filtre)
Mechanizmy zobrazovania	Predovšetkým grafické, Hlavný text, Integrácia grafiky a textu
Dodatočné údaje	Obsahuje „link“, žiadne dodatočné data

Tabuľka 2.1: Kategorizácia dashboardov podľa Stephena Fewa

2.2.1 Klasifikácia dashboardov podľa roly

Jedným z najužitočnejších spôsobov kategorizácie dashboardov, ktorej sa oplatí venovať pozornosť, je podľa roly. Táto kategória má najbližší vzťah k vizuálnemu dizajnu a je možné rozdeliť ju na nasledujúce 3 časti: strategické, analytické a prevádzkové účely.

- **Dashboards pre strategické účely**

Hlavné využitie týchto dashboardov je pri strategických rozhodnutiach. Poskytujú rýchly prehľad informácií potrebných pre ľudí, ktorý rozhodujú, ďalej sledovať stav a príležitosti v podnikaní. Dashboards tohto typu sú zamerané na meranie výkonu, ďalej môžu obsahovať aj odhady. Napriek tomu, že ďalšie podrobnosti ako kontext pre objasnenie dát, porovnanie alebo stručná história môžu byť výhodné, príliš veľa takýchto informácií môže od viesť pozornosť od primárnych a bezprostredných cieľov.

Ďalej nie sú potrebné aktuálne údaje v reálnom čase, ale stačia mesačné, týždenné alebo denné snímky. Taktiež vyhovuje statický spôsob zobrazenia dát bez interakcie.

- **Dashboardy na analytické účely**

Tieto dashboardy potrebujú iný prístup dizajnu. V týchto prípadoch informácie často využívajú väčší kontext, porovnania a rozsiahlejšiu históriu. Ďalej tiež profitujú zo statických snímok, ktoré sa nemenia neustále z jednej chvíli na druhú. Je veľmi dôležité dať používateľovi možnosť podrobnejšie preskúmať niektoré údaje alebo časové obdobie v prípade potreby. Z tohto plynie, že na rozdiel od strategických dashboardov, analytické by mali podporovať interakciu s užívateľom. Je to potrebné, nakoľko nestačí len zistiť následky, ale aj dôvody a analyzovať možné riešenia.

- **Dashboardy na prevádzkové účely**

Hlavný rozdiel od prvých dvoch typov je, že dáta sú dynamické. Pri sledovaní danej situácie je potrebné udržať ostražitosť na aktivity a udalosti, ktoré sa neustále menia a je potrebná možnosť okamžitej reakcie. Podobne ako u strategických dashboardov, zbytočné údaje a podrobnosti môžu mať negatívny vplyv hlavne v núdzových situáciách, keď sa očakáva okamžitá reakcia. Z toho plynie, že jednoduché grafické prvky sú uprednostnené. Nakoľko niekedy je nedostatok detailných informácií je užitočná interaktivita, ktorá umožňuje zobrazovať skryté detaily.

2.3 Efektívne médiá pre zobrazenie dát na dashboardoch

V tejto sekcii sa oboznámime s vhodnými zobrazovacími médiami k dosiahnutiu potenciálu dashboardov pre jasnú a rýchlu komunikáciu dát, ktoré Stephen Few zaradil do zbierky ideálnych médií. Upozorňuje na to, že časom sa táto zbierka pravdepodobne zmení, nakoľko sa musí prispôbiť k očakávaniam a zobrazovaným typom dát.

2.3.1 Zbierka ideálnych médií pre zobrazenie dát na dashboardoch

Predovšetkým je potrebné sa rozhodnúť, či je prospešné zobrazovať dané dáta nejakým grafickým elementom, jednoduchým textom alebo ich kombináciou. Je zbytočné komplikovať dashboardy s nejakým grafom v prípade, ak by bolo použitie textu jednoznačnejšie a jednoduchšie. Ďalej je potrebné rozmiestniť tieto prvky po dashboarde a v prípade výberu grafov sa rozhodnúť, ktorý je najvhodnejší. V tejto sekcii sa tiež zamierame na grafické elementy, ktoré slúžia na zobrazenie dát. Dve najdôležitejšie vlastnosti, ktoré musia spĺňať sú nasledovné: musí to byť najlepší spôsob zobrazenia daného typu dát a musí byť schopný splniť svoju rolu aj v omedzenom priestore. Celú zbierku je možné rozdeliť do niekoľkých kategórií: grafy, obrázky, ikony, text.

1. Grafy

Väčšina grafických elementov, používané na dashboardoch sú grafy. Najčastejšie typy dát zobrazených na dashboardoch sú početné. Stephen Few [5] vymenoval nasledujúce typy grafov:

- Bullet graphs
- Bar graphs (horizontal and vertical)
- Stacked bar graphs (horizontal and vertical)

- Combination bar and line graphs
- Line graphs
- Sparklines
- Box plots
- Scatter plots
- Treemaps

2. Obrázky

Všeobecne použitie obrázkov na dashboardoch môže byť užitočné v niektorých prípadoch, ako napr. mapa budovy pre údržbára. Z pohľadu dashboardov, ktoré sú používané v podnikaní, však nie sú potrebné.

3. Ikony

Sú to jednoduché obrázky, ktoré komunikujú jasný a jednoduchý odkaz. Na dashboardoch je ich potrebné len niekoľko, najužitočnejšie sú nasledujúce tri: upozornenia, smer (hore/dole), stav (on/off). Tieto ikony je vidieť na obrázku [2.2](#).



Obr. 2.2: Najčastejšie používané ikony.

4. Text

Niektoré informácie sa lepšie vyjadrujú textom ako graficky. Príkladom je zobrazenie nejakej číselnej informácie bez porovnania, na čo je ideálne použiť jednoduchý textový box.

2.4 Časté problémy v návrhu dashboardov

Základnou výzvou dizajnu dashboardov je potreba dostať veľké množstvo informácií do malého priestoru tak, aby zobrazenie týchto informácií bolo jednoznačné a zrozumiteľné. Žiaľ, je ľahké nájsť veľa príkladov dashboardov na stránkach samotných dodávateľov softvéru pre tvorbu dashboardov, ktoré nie vždy majú kvalitné zobrazenie dát a malo by sa im vyhnúť. V tejto podkapitole uvádzame niekoľko problémových obrázkov a popis týchto častých nedostatkov.

1. Fragmentovanie dashboardov

Informácie, ktoré sa zobrazujú na dashboardoch, môžu byť fragmentované nasledujúcimi spôsobmi: rozdelené na oddelené obrazovky alebo pomocou rolovania. Príklady takýchto dashboardov vidieť na obrázku [2.3](#), ktoré ukazujú fragmentáciu. S porušením dôležitých súvislostí súvisí problém krátkodobej pamäti, ktoré je popísané v sekcii [4.1](#)



Obr. 2.3: Príklad problémových dashboardov. Dashboard by sa mal zmestiť na jednu obrazovku počítača. Je potrebné vybrať daný produkt kliknutím alebo rolovať po obrazovke. [5]

2. Výber nevhodných zobrazovacích médií

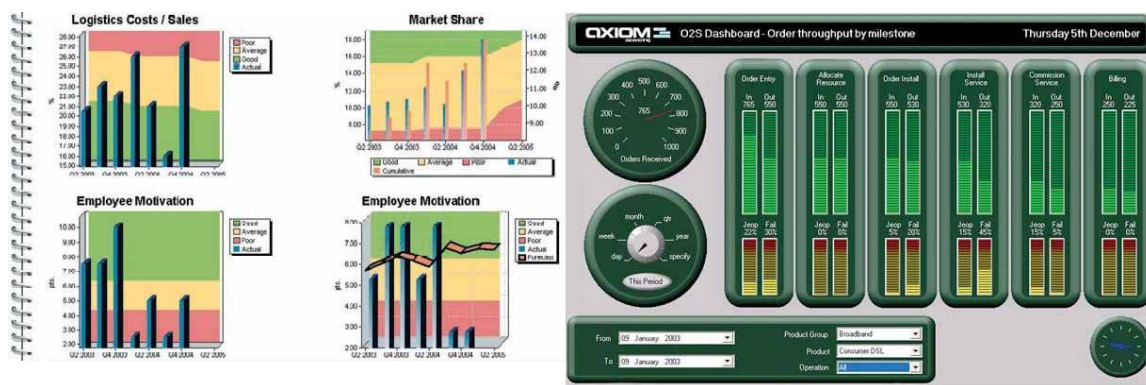
Výber nevhodného zobrazovacieho média je jedna z najbežnejších návrhových nedostatkov, a to nielen v dashboardoch, ale aj vo všetkých formách prezentácie údajov. Napríklad použitie grafu, keď tabuľka čísiel by bola vhodnejšia a naopak. Príklad nevhodného kruhového grafu a presnejšieho vodorovného stĺpcového grafu je vidieť na obrázku 2.4.



Obr. 2.4: Obrázok zobrazuje bežný problém s kruhovými grafmi, veľkosť jednotlivých častí neodpovedá daným percentám. [5]

3. Nepotrebná a zbytočná dekorácia displeja

Bežným problémom je množstvo zbytočných výzdob. Hocijaké prínosy takýchto dizajnov nie sú dlhodobé. Celkovo dashboardy by mali uprednostniť použiteľnosť. Ak divák musí spracovať vizuálny obsah dashboardu, aj keď trvá iba niekoľko sekúnd, aby sa dostal k údajom, sú nevhodné pre časté použitie. Takéto dashboardy je vidieť na obrázku 2.5.



Obr. 2.5: Tieto dashboards sú príkladom zbytočnej dekorácie, čo má za následok nepoužiteľnosť. [5]

4. Problém rozloženia

Najdôležitejší problém z pohľadu tejto práce, týkajúci sa vizualizácie údajov, je nevhodné rozloženie grafických elementov. Nepriehľadné usporiadanie spôsobuje, že užívateľ stratí zbytočne čas, kým získa prehľad daného rozhrania. Takéto dashboards nie sú vhodné pre dlhodobé používanie. Predísť tomuto problému je možné použitím metrick analýzy rozloženia, ktoré sú popísané v kapitole 3.

5. Ďalšie možné problémy

- Veľa podrobností, ktoré sú vyjadrené príliš presne (napríklad namiesto 3 848 305,93 USD použiť výraz 3,848,305 dolárov, alebo ak je možné ešte kratší výraz 3,8 milión dolárov)
- Zvýraznenie dôležitých údajov neúčinne alebo vôbec
- Navrhovanie neatraktívnych dashboardov

Kapitola 3

Metriky pre meranie vlastností užívateľských rozhraní

Obrazovka by mala vhodne prezentovať informácie a tiež zabezpečiť interakciu so systémom. Ako bolo ukázané v sekcii 4, grafické elementy užívateľského rozhrania by mali byť na obrazovke vhodne rozmiestnené. Dodržanie doporučených aspektov rozloženia vedie k lepšej orientácii užívateľa v užívateľskom rozhraní. Znalosť úloh a schopností užívateľov je kľúčom k navrhnutiu efektívnych obrazoviek, pričom objektívna a automatizovaná metrika návrhu obrazovky je základnou pomôckou.

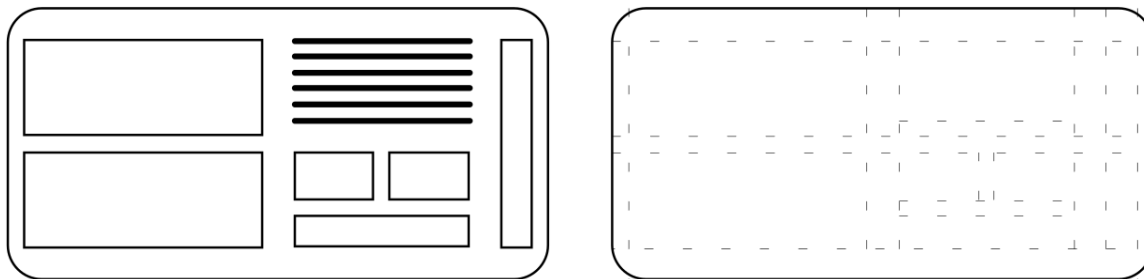
V tejto kapitole sa zameriame na popis metrík, ktoré vyhodnocujú rozmiestnenie grafických elementov v užívateľskom rozhraní, respektíve na obrazovke.

3.1 Popis metrík

- **Kvalitatívny popis metrík**

Podľa publikácie *Visual Techniques for Traditional and Multimedia Layouts* [16] návrhári užívateľských rozhraní majú dve hlavné úlohy pri návrhu. Prvá úloha je výber vhodných grafických prvkov/objektov, tiež nazvané v tejto publikácii *Interaction Object* (IO). Druhou úlohou je vytvoriť rozloženie týchto objektov podľa dôležitosti.

Rozloženia užívateľských rozhraní sa skladajú zo sady obdĺžnikov, ktoré ohraničujú grafické objekty (regióny). Mriežka rozloženia sa potom skladá z paralelných horizontálnych a vertikálnych čiar. Týmto rozdeľujú objekty v obrázku na jednotky, ktoré majú vizuálnu a koncepcnú integritu. Tieto dva popisy rozloženia užívateľských rozhraní je vidieť na obrázku 3.1.



Obr. 3.1: Rozloženie užívateľského rozhrania a jeho mriežka rozloženia. [16]

Táto publikácia ďalej obsahuje kvalitatívny popis metrík, vhodných na analýzu rozloženia. Tieto metriky vyhodnocujú vlastnosti objektov zobrazených v užívateľských rozhraniach. Sú vhodné napr. na analýzu pozície a veľkosti objektov. Podľa týchto kvalitatívnych popisov metrík je náročné vytvoriť algoritmy a programy nakoľko sú popísané slovné a pomocou obrázkov a nie sú presne zadefinované matematickým postupom. Autori tejto publikácie navrhujú rozdelenie vizuálnych techník do 5 skupín.

1. **fyzikálne techniky** – rovnováha, symetria, pravidelnosť, zosúladenie, pomer a horizontálnosť
2. **kompozičné techniky** – jednoduchosť, hospodárnosť, podhodnotenie, neutralita, singularita, pozitívnosť a transparentnosť;
3. **asociačné a disociačné techniky** – jednota, rozdelenie, zoskupovanie a šetrenie;
4. **poradové techniky** – konzistencia, predvídateľnosť, postupnosť a kontinuita;
5. **fotografické techniky** – ostrosť, zaoblenosť, stabilita, vyrovnanie, aktivita, jemnosť, reprezentácia, realismus a rovinnosť.

- **Kvantitatívny popis metrík**

Kvantitatívne popisy metrík boli vytvorené neskôr, ktoré sú popísané v niekoľkých publikáciách. [11, 12, 13] Tieto popisy obsahujú aj matematické vyjadrenie týchto metrík, ktoré je možné ľahko previesť na algoritmy. Príklad takéhoto popisu znázorňujú vzorce 3.1 až 3.4, pre meranie vyváženosti.

$$BM = 1 - \frac{|BM_{vertical}| + |BM_{horizontal}|}{2} \in [0, 1] \quad (3.1)$$

$$BM_{vertical} = \frac{W_L - W_R}{\max(|W_L|, |W_R|)} \quad (3.2)$$

$$BM_{horizontal} = \frac{W_T - W_B}{\max(|W_T|, |W_B|)} \quad (3.3)$$

$$w_j = \sum_i^{n_j} a_{ij} d_{ij} j = L, R, T, B \quad (3.4)$$

Kde:

$BM_{vertical}$, $BM_{horizontal}$ – vertikálna resp. horizontálna vyváženosť;

L, R, T, B – stoja za *left*, *right*, *top* a *bottom*;

w_j – celková váha strany j ;

a_{ij} – obsah objektu i na strane j ;

d_{ij} – vzdialenosť is the distance between the central lines of the object and the frame;

a n_j – celkový počet objektov na strane j .

Podľa článku *Application of an aesthetic evaluation model to data entry screens* [12] je možné identifikovať 13 dôležitých charakteristík esteticky príjemných objektov, ktoré sú vysvetlené ďalej v tejto kapitole a príklady zobrazené na obrázku 3.2.

1. **Meranie vyváženosti** (Measure of balance)

Vyváženosť obrazu môžeme definovať ako rozloženie optickej hmotnosti obrazu, čo znamená, že niektoré objekty vyzerajú „ťažšie“ ako iné. Väčšie objekty sú ťažšie, kým menšie objekty sú ľahšie. Vyváženosť obrazu je dosiahnutá rovnakou hmotnosťou objektu obrazovky na ľavej a pravej, hornej a spodnej časti strany.

2. **Meranie rovnováhy** (Measure of equilibrium)

Rovnováhu obrazovky môžeme definovať ako stabilizáciu uprostred ťažiska. Rovnováha na obrazovke sa dosiahne centrovaním usporiadania objektov. Rovnováha je do určitej miery spojená s vyváženosťou. Z pohľadu rovnováhy je dôležité vizuálne centrum, oproti tomu z pohľadu vyváženosti je dôležitá vizuálna hmotnosť.

3. **Meranie symetrie** (Measure of symmetry)

Symetriu obrazu môžeme definovať ako osovú duplikáciu, ktorá je dosiahnutá ako replikácia prvkov z jednej strany osi na druhú stranu osi. Vertikálna symetria sa vzťahuje na vyváženosť usporiadania ekvivalentných prvkov okolo vertikálnej osi a horizontálna symetria okolo horizontálnej osi.

4. **Meranie sekvencie** (Measure of sequence)

Sekvenciu v dizajne obrazu môžeme definovať ako usporiadanie objektov takým spôsobom, aby uľahčovali pohyb oka nad zobrazenými informáciami. Obecný pohyb oka, trénovaný čítaním, začína v ľavom hornom rohu a pohybuje sa zľava doprava tam a späť cez displej, kým sa dostane do pravého dolného rohu. Psychológovia zistili, že niektoré prvky priťahujú oko viacej, ako napr. veľkosť, z čoho plynie, že oko prechádza z veľkých objektov na malé objekty.

5. **Meranie súdržnosti** (Measure of cohesion)

V dizajne obrazovky dodržanie pomeru strán naznačuje súdržnosť. Pojem pomer strán odkazuje na vzťah medzi šírkou a výškou obrazovky. Typická veľkosť papiera je vyššie ako širšie, zatiaľ čo opak platí pre typické displeje (najčastejšie 16:9).

6. **Meranie jednoty** (Measure of unity)

Jednota je súdržnosť, alebo celok prvkov, ktoré vizuálne vyzerajú ako jeden celok. V jednote prvky sa zdajú, ako by patrili spolu, až natolko, že sú vnímané ako jedna vec. Jednotnosť v dizajne obrazovky sa dosahuje použitím podobných veľkostí prvkov a ponechávaním menších medzier medzi prvkami obrazovky, ako je medzera okrajov.

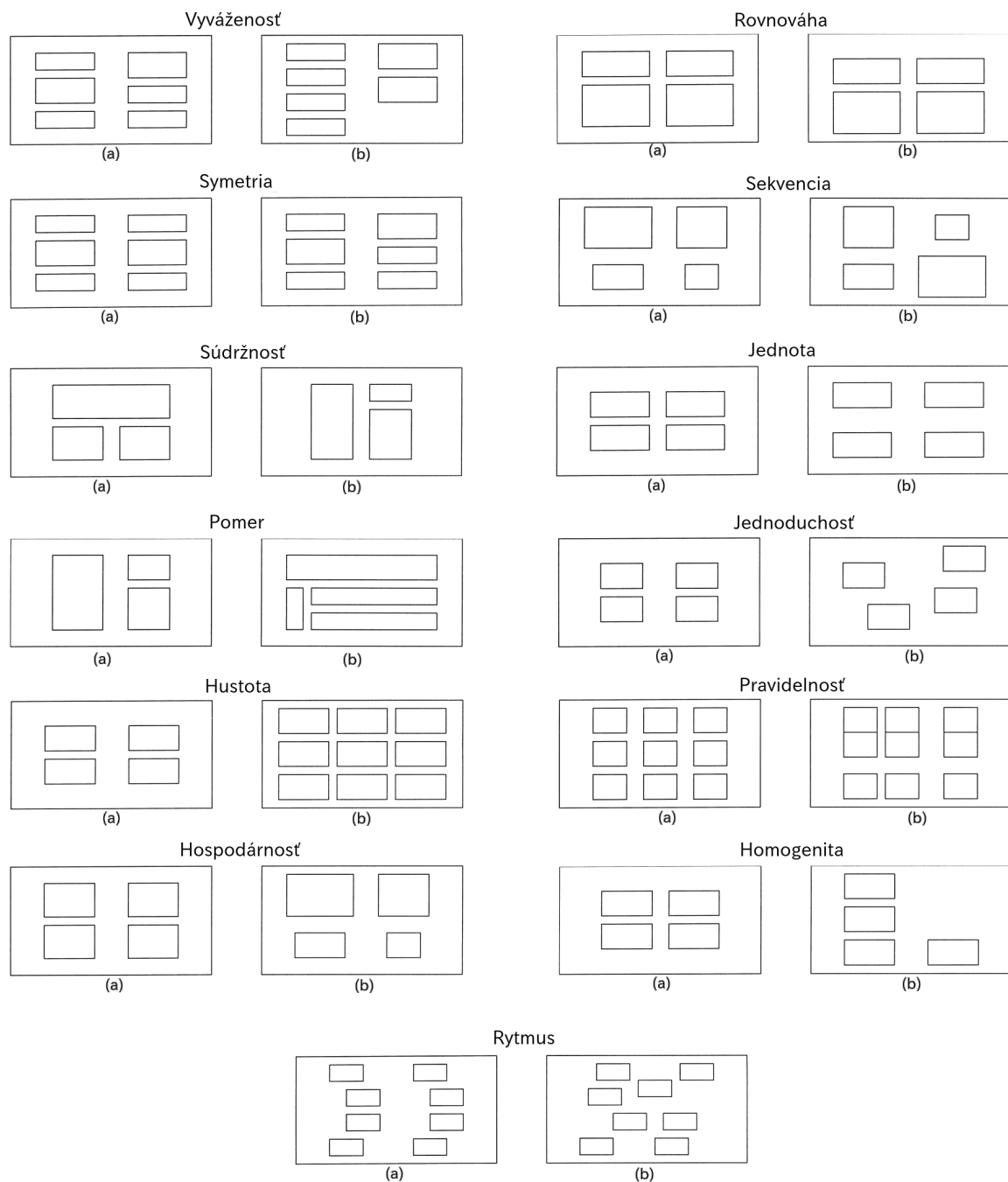
7. **Meranie pomeru** (Measure of proportion)

Pomer sa zaoberá vzťahom veľkosti jedného bloku k inému bloku. Pri dizajne obrazovky je potrebné zvážiť estetické pomery veľkých komponentov obrazovky, vrátane okien a skupín údajov a textu.

8. **Meranie jednoduchosti** (Measure of simplicity)

Jednoduchosť je primeranosť a jednotnosť formy, kombinácia prvkov, ktorá vedie k ľahkému pochopeniu významu vzoru. Jednoduchosť dizajnu obrazovky je dosiahnutá optimalizáciou počtu prvkov na obrazovke a minimalizáciou bodov zarovnania.

9. **Meranie hustoty** (Measure of density)
Hustota je princíp dizajnu, ktorá zahŕňa optimalizáciu obsadených oblastí. Vo všeobecnosti ak je využito príliš málo miesta, tak obrazovka vyzerá byť prázdna, ale v opačnom prípade zase preplnená. Preto sa obmedzuje hustota úrovni využitia obrazovky na optimálne percento.
10. **Meranie pravidelnosti** (Measure of regularity)
Pravidelnosť je jednotnosť prvkov, ktorá je založená na nejakom princípe alebo pláne. Pravidelnosť dizajnu obrazovky sa dosahuje stanovením štandardných a konzistentných rozmiestnení horizontálnych a vertikálnych vyrovnávacích bodov pre prvky obrazovky, a minimalizovanie bodov vyrovnania. Kým obe metríky, pravidelnosť a jednoduchosť, súvisia s počtom horizontálnych a vertikálnych vyrovnávacích bodov, pravidelnosť, na rozdiel od jednoduchosti, je menej citlivá na počet prvkov na obrazovke.
11. **Meranie hospodárnosti** (Measure of economy)
Ekonomia v dizajne obrazovky je používanie prvkov opatrným a diskrétnym spôsobom na zobrazenie dát najjednoduchším spôsobom ako sa len dá. Ekonomika sa dosahuje s použitím čím najmenšieho počtu rozmeru prvkov ako je len možné. Doporučuje sa, aby aspoň elementy v rámci skupiny boli podobné.
12. **Meranie homogeneity** (Measure of homogeneity)
Homogénnosť je podľa definície merateľnosť rovnomerného rozdelenia objektov medzi kvadrantmi.
13. **Meranie rytmu** (Measure of rhythm)
Rytmus v dizajne sa vzťahuje na pravidelné zmeny vzorov v prvkoch. Rytmus sa dosahuje zmenou usporiadania, rozmeru, počtu a formy prvkov. Miera, do ktorého je rytmus zavedený, závisí od zložitosti skupiny prvkov (počet, rozdielnosť).



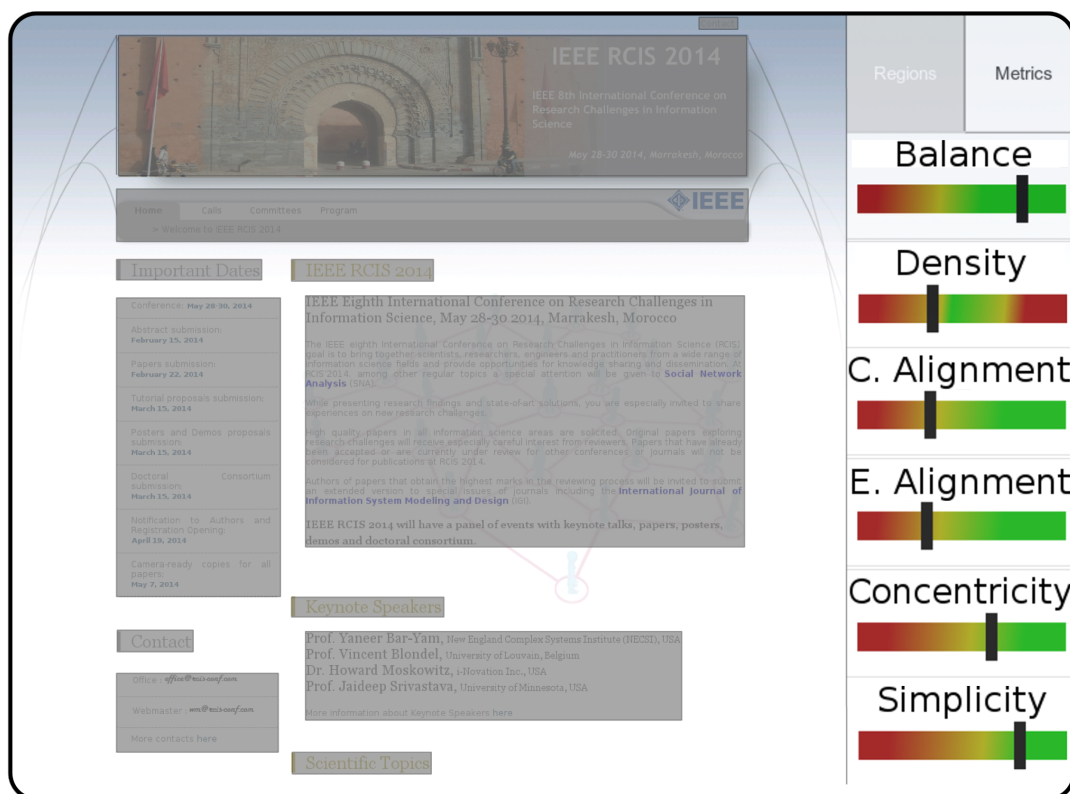
Obr. 3.2: Obrázky (a) zobrazujú optimálne navrhnutú obrazovku a obrázky (b) zobrazujú nevhodne navrhnutú obrazovku podľa danej metriky. [12]

3.2 Využitie metrík v súčasnosti

Vyššie uvedené metriky sa využívajú aj v súčasnosti. Príkladom môže byť publikácia *Towards an Evaluation of Graphical User Interfaces Aesthetics based on Metrics* [18], v ktorej autori navrhujú metódu analýzy a vyhodnotenia užívateľských rozhraní. Je to jednoduchý a iteratívny spôsobom, ktorý sa skladá zo štyroch krokov, podľa ktorých bol aj pomenovaný *INFRARED*.

1. Select **IN**ter**F**ace - výber užívateľských rozhraní na analýzu
2. Draw **R**egions - označenie vizuálne dominantných regiónov
3. Analyze **A**esthetics - vyhodnotenie užívateľských rozhraní podľa metrík na základe regiónov z kroku 2.
4. **RE**Design UI - prepracovanie užívateľského rozhrania podľa výsledkov kroku 3.

Na základe metódy INFRARED vyvinuli nástroj QUESTIM, ktorý je schopný spracovať url webovej stránky alebo obrázok užívateľského rozhrania nahraný z disku. Od užívateľa sa očakáva, aby označil regióny, s ktorými pracujú metriky. Výsledky vyhodnotenia sú zobrazené na pravej strane nástroja ako vidieť na obrázku 3.3. Pre znázornenie vhodnosti sa používa farebná škála, kde odtieň zelenej označuje dobrý návrh podľa danej metriky a červená farba nevhodný návrh.



Obr. 3.3: Nástroj analýzy užívateľských rozhraní Questim. [18]

Jedným z možných vylepšení zmieneného prístupu vyhodnocovania je, aby bolo možné vykonať túto segmentáciu automaticky a umožniť užívateľovi upravovať výsledné regióny (veľkosť, pozícia, pridať/odstrániť regióny), čím sa zaoberá táto práca.

Kapitola 4

Vizuálne vnímanie objektov v užívateľských rozhraniach

Objekty v obraze môže každý človek rozpoznávať mierne odlišným spôsobom. Z toho vzniká problém pri použití metrík hodnotiacich vlastností užívateľských rozhraní. Pri použití nástroju 3.2 rôznymi užívateľmi, ktorý na tom istom obrázku označia regióny odlišne, metriky budú mať rôzne výsledky aj keď sa jedná o jeden a ten istý obrázok. Vzniká nutnosť pochopiť, ako ľudia vnímajú objekty. V tejto kapitole sa dozvieme o faktoroch vizuálneho vnímania človeka, ktoré ovplyvňujú návrh dashboardov z rôznych pohľadov. Sú to nasledovné faktory: obmedzená krátkodobá pamäť, vizuálne kódovanie pre rýchle vnímanie a Gestalt princípy.

Vizuálne vnímanie označuje schopnosť interpretovať okolité prostredie pomocou svetla vo viditeľnom spektre. Približne 70% zmyslových receptorov v tele sa venujú práve zraku a spracovaniu obrazu.

Ďalej tieto princípy môžu vylepšiť návrh dashboardov, nakoľko niektoré výzvy dobrého dashboardu, ako vhodný spôsob zvýraznenia najdôležitejších dát, môžu byť prekonané múdrou aplikáciou niektorých z týchto princíпов.

4.1 Obmedzenia krátkodobej pamäte

K tomu, aby bolo jasné, aký má vplyv krátkodobá pamäť na dizajn dashboardov, najprv je potrebné porozumieť tomu, ako funguje ľudský mozog a zrak. V skutočnosti oči slúžia iba ako „senzor“ svetla. Informácie, získané očami sú ďalej prevedené na elektrické impulzy neurónmi, ktoré sú ďalej prenesené do mozgu a spracované. Ale iba na malú časť toho čo vníma oko je schopný sa sústrediť človek a tieto časti sú uložené do pamäte človeka. Stephen Few sa zmienil o existencii troch rôznych typov pamäte [5]:

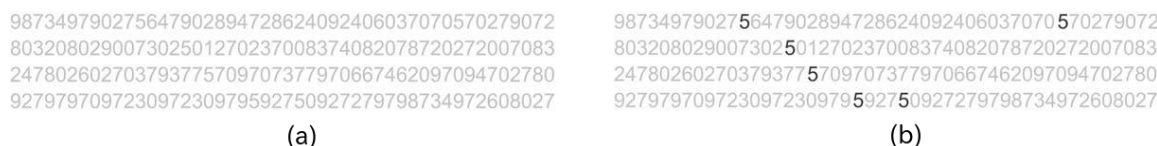
1. Podvedomá pamäť (podobné počítačovým registrom)
2. Krátkodobá pamäť (podobné operačnej pamäti)
3. Dlhodobá pamäť

Pri vnímaní a spracovaní obrazu mozog najprv detekuje objekty iba podľa niektorých atribút (farba, tvar..), čo sa deje v podvedomí. S týmto je spojené podvedomé spracovanie obrazu a podvedomá pamäť. Ďalej sa spracuje obraz pozorne a sekvenčne. Takéto

spracovanie je oveľa pomalšie a prebieha vedome. Príklad na porovnanie týchto spôsobov spracovania je možné vidieť na obrázku 4.1, ktorý znázorňuje vyhľadávanie konkrétneho čísla v zozname.

- V prvom prípade (obrázok a) sa v podvedomí rozpozná iba zoznam objektov s rovnakou farbou. Ľudský mozog v krátkodobej pamäti je schopný uchovať len malé množstvo informácií. Pri sekvenčnom spracovaní týchto objektov, človek je schopný sústrediť sa súčasne iba na 3 - 9 objektov. Počet týchto objektov odpovedá veľkosti krátkodobej pamäti jednotlivca.
- V druhom prípade (obrázok b) sa v podvedomí rozpoznávajú dva typy objektov podľa farby. Pri vedomom vyhľadávaní sa rýchlejšie nájdu hľadané čísla ktoré boli rozpoznané v podvedomí.

Tieto vizuálne vlastnosti je možné použiť pri dizajne dashboardov na zoskupenie alebo zvýraznenie informácií.

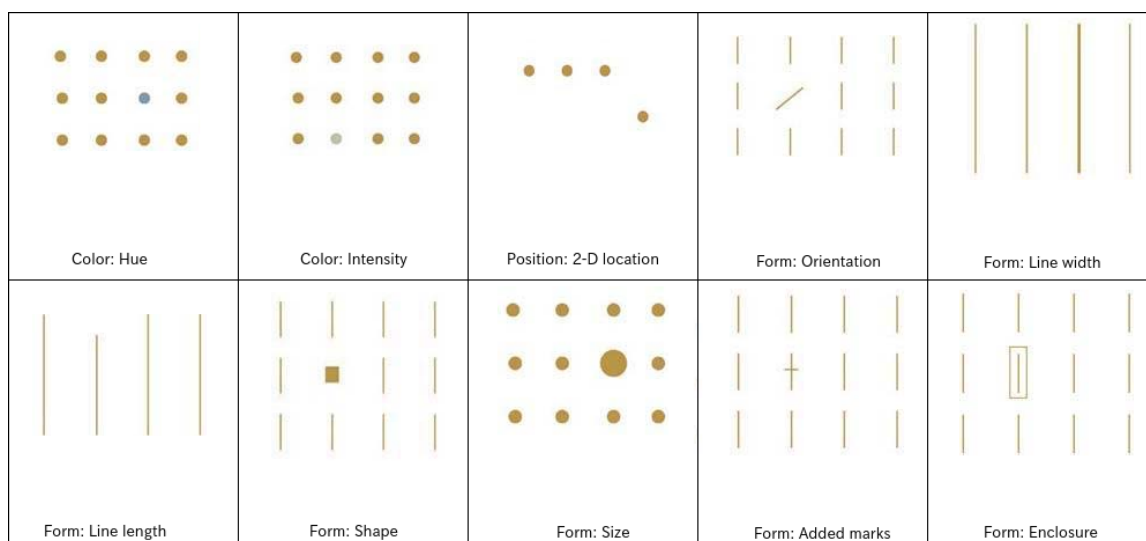


Obr. 4.1: Obrázok (a) predstavuje zoznam bez vyznačenia a obrázok (b) zobrazuje zoznam s vyznačením konkrétneho čísla. [5]

Obmedzená kapacita krátkodobej pamäte je dôvodom, prečo informácie, ktoré patria k sebe, by nikdy nemali byť rozdelené do viacerých dashboardov. Ideálnym príkladom je potreba rolovať cez obrazovku. Ak dáta, ktoré patria k sebe sú rozdelené, pozorovateľ sa môže dostať do situácie, keď musí prerolovať značnú časť obrazovky viackrát, aby vedel spracovať informácie, s čím stratí zbytočne čas a energiu. Pokiaľ všetko potrebné zostáva v rámci očného rozpätia na jedinom dashboarde, je možné rýchlo vymieňať informácie z krátkodobej pamäte, a to s rýchlosťou blesku.

4.2 Atribúty podvedomého rozpoznávania objektov

Autor knihy Information Visualization: Perception for Design [17], Colin Ware navrhol kategorizáciu podvedomého rozpoznávania objektov podľa nasledujúcich atribút: farba, tvar, poloha a pohyb. Na základe týchto atribút človek podvedome odlišuje objekty. Podvedomé rozlišovanie prebieha bez toho, aby sa na to človek musel sústrediť. Dôležité prvky, z pohľadu na dizajn dashboardov, z tejto kategórie je vidieť na obrázku 4.2. Jediným predstaviteľom pohybu z tejto kategórie je blikanie. Nejaká vlastnosť niektorého objektu, ako napríklad farba, neustále mení svoju hodnotu medzi dvoma hodnotami, alebo celý objekt opakovane zmizne a znovu sa objaví.



Obr. 4.2: Kategorizácia prvej pozornosti, dôležitých z pohľadu dizajnu dashboardov. [5]

1. Farba

Ľudia nevnímajú farbu absolútnym spôsobom, čo má za následkom, že vnímaná farba je dramaticky ovplyvnená svojím okolím, hlavne odtieňom a jasom. Na obrázku 4.3 je vidieť príklady na tento jav. Na obrázku (a) môžeme pozorovať 5 rovnakých štvorcíkov, kde 4 sú umiestnené pred pozadím s jasom čiernej farby s farebným prechodom od bielej (100) k čiernej (0), a jeden samostatne s bielou farbou pozadia. Kým jas pozadia klesá z ľavej strany, štvorčeky rovnakej farby umiestnené pred týmto pozadím vyzerajú tmavšie na ľavej strane. Podobný prípad je vidieť na obrázku (b) keď viditeľnosť textu ovplyvňuje odtieň farby. Slovo „Text“ má rovnakú farbu na oboch stranách, predsa nie sú rovnako viditeľné pred červeným a pred modrým pozadím. Pri dizajne dashboardov je potrebné venovať veľkú pozornosť farbám, odtieňom a jasom, nakoľko okrem čitateľnosti informácií na dashboardoch je potrebné, aby jednotlivé elementy a časti dashboardov vyzerali rovnako alebo podobne ak sa to vyžaduje, alebo v inom prípade naopak aby boli dobre rozlíšiteľné.



Obr. 4.3: Obrázky zobrazujú vplyv pozadia na farbu konkrétneho objektu. [5]

2. Tvar

Podľa tvaru poznáme niekoľko možností, a to:

- Najlepším príkladom orientácie je typ písma kurzíva. Funkciou tohto písma je zvýrazniť daný text, ale takéto písmo sa ťažšie číta.

- Vlastnosť dĺžka čiary je užitočná na zakódovanie spočetných dát v stĺpcových grafoch.
- Šírka čiary vo všeobecnosti pri ohraničených objektoch môže upútať viac pozornosti.
- Veľkosť objektov dashboardov môže byť užitočná na vizuálne vyhodnotenie podľa dôležitosti. Napríklad väčšie titulky, tabuľky, grafy alebo iné grafické elementy zvýrazňuje väčšiu dôležitosť zobrazených dát.
- Pridanie značky v dashboarde vedľa informácií je možné použiť na zvýraznenie týchto informácií.
- Poslednou vlastnosťou je ohraničenie, čo je silným prostriedkom zoskupenia dát alebo zvýraznenie oblasti nasledujúcimi spôsobmi: ohraničením čiarou alebo s inou farbou pozadia.

3. Poloha

Poloha v prvotnej pozornosti je primárny prostriedok, ktorý používame na zobrazenie početných údajov v grafoch (napr. umiestnenie údajov vo vzťahu k číselnej stupnici).

4. Pohyb

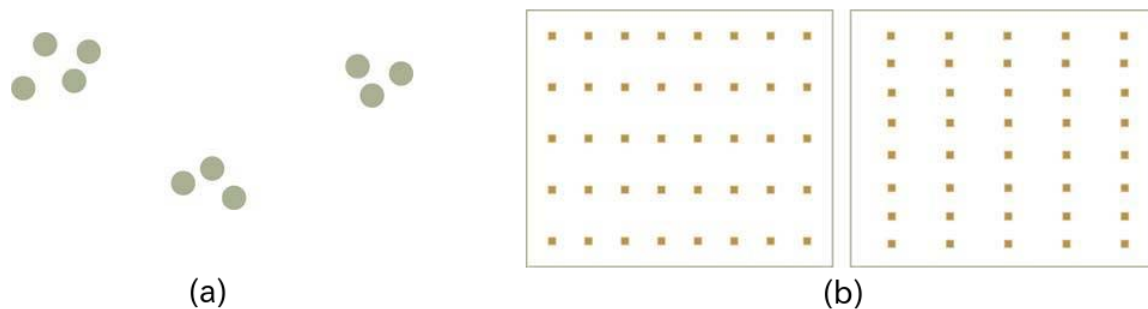
Ako bolo popísané vyššie, blikanie vznikne zmenou nejakej vlastnosti farby alebo zmiznutím objektu a znovu objavením. Takéto blikanie má veľkú silu upútať pozornosť, ako pri písaní kurzor aktuálnej pozície. Obecne blikajúce objekty a čiary sú veľmi nepríjemné, a preto by sme sa im mali vyhýbať. Na druhej strane, ak sa jedná o dashboardy, ktoré zobrazujú aktuálne, neustále meniace sa dáta, tento jav môže byť užitočným a pomôcť pozorovateľovi rýchlejšie zachytiť zmeny.

4.3 Gestalt princípy vizuálneho vnímania

Gestalt princípy pochádzajú z Nemecka (nemecký výraz *gestalt* znamená „vzor“) a snažia sa vysvetliť, ako vnímame vzory, formy a organizáciu v tom, čo vidíme. Sú to pravidlá, ktoré popisujú ako ľudia podvedomo zhlukujú objekty do väčších celkov a vnímajú ich ako celok. Súvisia s atribútami podvedomého vnímania, ale aj s omedzenou veľkosťou krátkodobej pamäti. (Človek nie je schopný sa sústrediť na všetky objekty, podvedome zjednodušuje obraz.) Nemecký vedci vytvorili zbierku týchto princípov. Z pohľadu dizajnu dashboardov, najdôležitejšie sú nasledujúce: blízkosť, podobnosť, ohraničenie (enclosure), uzavretie (closure), kontinuita, pripojenie.

4.3.1 Princíp blízkosti

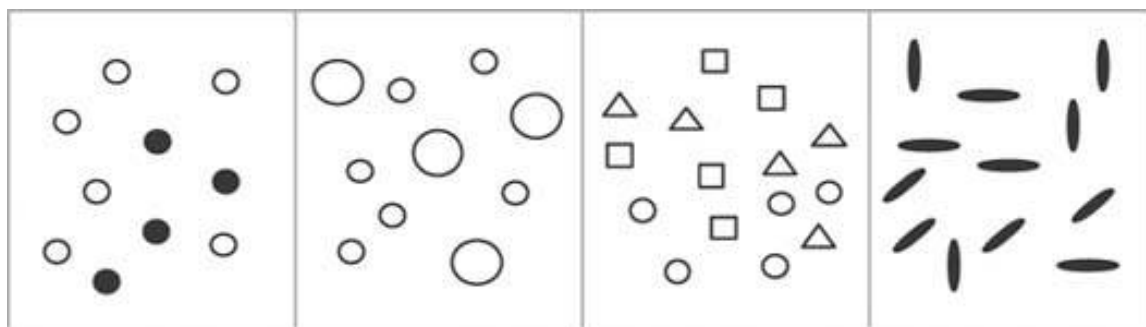
Objekty, ktoré sa nachádzajú blízko seba vnímame akoby patrili do jednej skupiny. Tento princíp môže mať vplyv na vnímanie smeru usporiadania objektov na danom obrázku. Obrázok 4.4 znázorňuje oba prípady.



Obr. 4.4: Obrázok (a) predstavuje zoskupenie objektov a obrázok (b) zobrazuje vplyv princípu blízkosti na smer usporiadania objektov. [5]

4.3.2 Princíp podobnosti

Ľudia majú tendenciu zoskupovať objekty, ktoré majú podobnú farbu, veľkosť, tvar alebo orientáciu. Obrázok 4.5 ilustruje túto tendenciu.

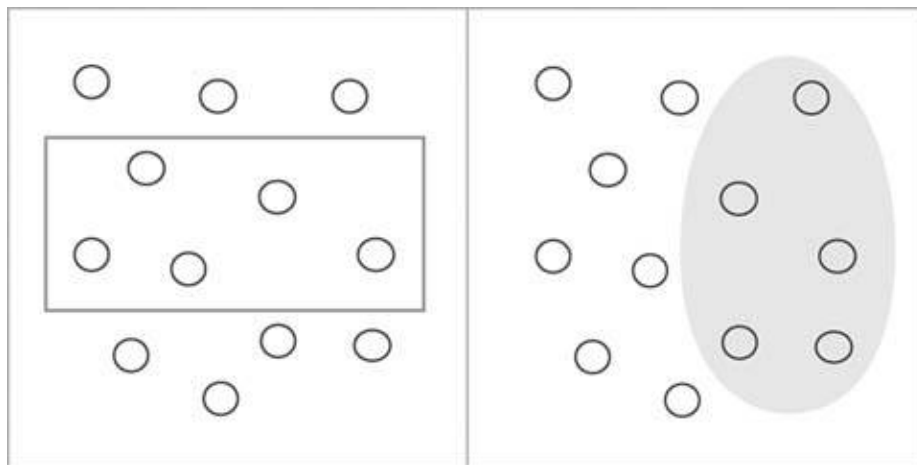


Obr. 4.5: Obrázky zobrazujú zoskupenie objektov podľa farby, veľkosti, tvaru a orientácie. [5]

V dizajne dashboardu sa tento princíp využíva na ľahšie rozlíšenie častí, ktoré nepatria, alebo naopak patria, k sebe. Konkrétny príklad je zobrazenie dát, ktoré sa vyskytujú vo viacerých grafoch, jednou farbou.

4.3.3 Princíp uzavretia (enclosure)

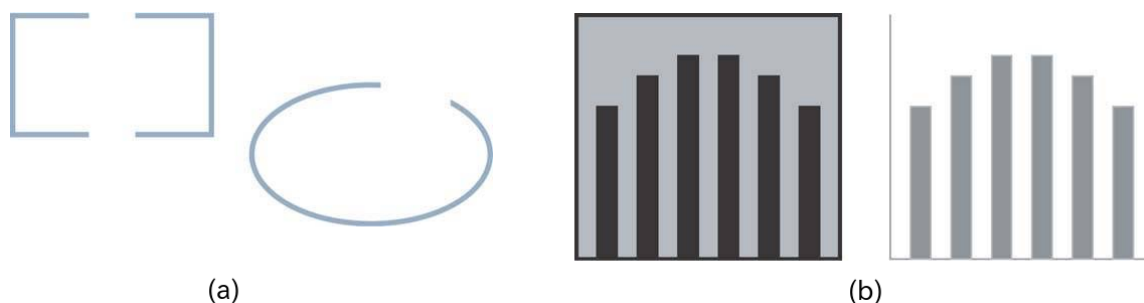
Objekty ohraničené rámčekom alebo spoločnou farbou pozadia sú vnímané ako jedna skupina. Môžeme to vidieť na obrázku 4.6. Aj keď počet a usporiadanie objektov je rovnaký, kvôli rámčeku a pozadiu vyzerajú ako dve totálne odlišné sady objektov.



Obr. 4.6: Obrázok (a) zobrazuje zoskupenie objektov rámčekom a obrázok (b) zobrazuje zoskupenie objektov spoločnou farbou pozadia.[5]

4.3.4 Princíp uzavretia (closure)

Ludia majú tendenciu vnímať objekty, ktoré majú chýbajúce časti, ako celé. Príklad tohto môžeme vidieť na obrázku 4.7(a) kde namiesto 3 nesúvislých čiar vidíme obdĺžnik a elipsu. Niektoré grafické prvky, ako grafy ak sú ohraničené rámčekom, vnímame ich akoby boli jeho hodnoty tiež ohraničené. Ak pravá a horná časť je vynechaná, grafický prvok je stále vnímaný ako ohraničená jednotka, ale už nevnímame podľa rámčeka horný a pravý limit dát. Toto je vidieť na obrázku 4.7(b).



Obr. 4.7: Obrázok (a) zobrazuje 3 čiary, ktoré sú vnímané ako geometrické tvary a obrázok (b) znázorňuje vplyv rámčeka na vnímanie ohraničenia grafu.[5]

4.3.5 Princíp kontinuity

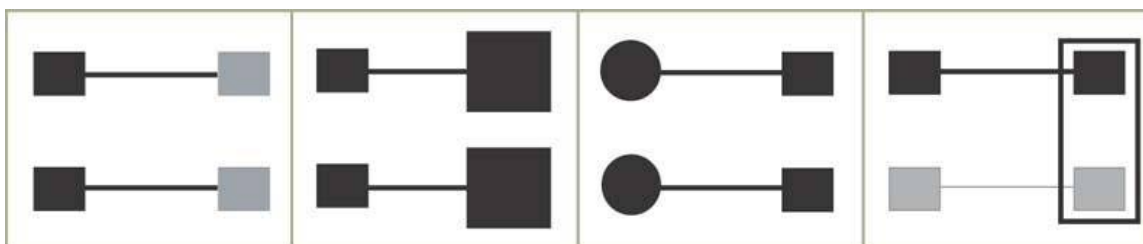
Základom tohto princípu je, ako oči vnímajú prechod z jedného objektu na ďalší. Jednotlivé časti sú vnímané akoby patrili k sebe a tvorili jeden objekt. Príkladom je obrázok 4.8, ktorý zobrazuje niekoľko čiar a bodiek. Prerušovanú čiaru vnímame ako jednu dlhú čiaru, a body ako dve čiary, ktoré sa navzájom križujú.



Obr. 4.8: Obrázok znázorňuje sadu čiar [5] a bodiek¹, ktoré sú vnímané ako súvislé čiar.

4.3.6 Princíp pripojenia

Objekty blízko k sebe sú často vnímané akoby patrili do jednej skupiny, ako je popísané v sekcii 4.3.1, a zobrazené na obrázku 4.4. Z obrázku 4.9 plynie, že blízkosť a spoločné vlastnosti (ako farba, veľkosť, tvar) vytvárajú slabší vzťah ako spojenie čiarou a najsilnejší vzťah vytvára ohraničenie.



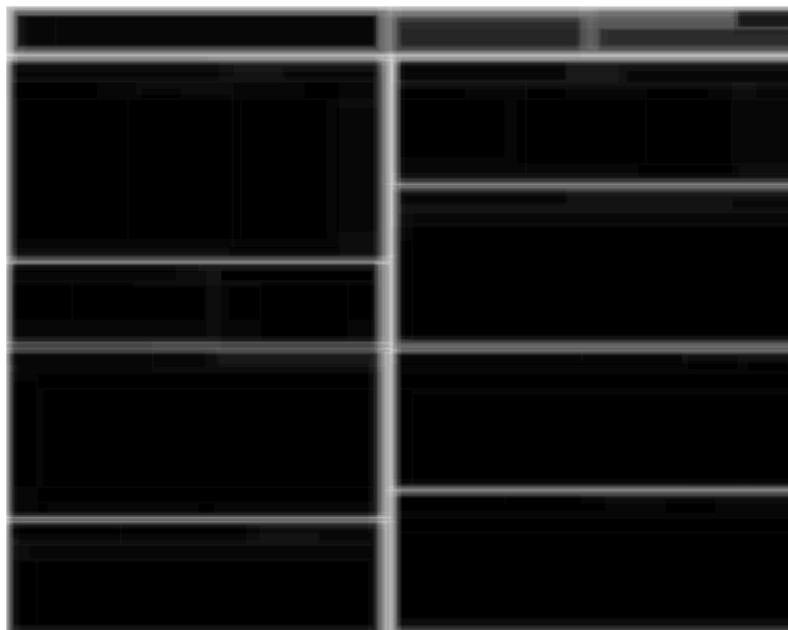
Obr. 4.9: Obrázky znázorňujúce odlišnosti zoskupení objektov spájaním podľa spoločných vlastností a ohraničením. [5]

4.4 Výskum rozpoznávania objektov

Na VUT FIT v Brne bol prevedený výskum², ktorý skúmal, ako ľudia zhlukujú objekty do väčších celkov. Študenti sa najprv oboznámili s nástrojom pre vyhodnotenie dashboardov. Za úlohu mali vyznačiť regióny reprezentujúce logické časti dashboardov, ktoré podľa ich názoru vizuálne vytvárali samostatný celok oddeliteľný od ostatných grafických celkov. Jednalo sa o jednotlivé grafy, prípadne menu, tlačítka, nadpisy, a podobne. Získané výsledky sa zobrazili pomocou bitmap vizualizujúce pravdepodobnosti výskumu regiónov (tzv. heatmapa). Heatmapa sa tvorí prevedením regiónov z XML výsledkov na matice s hodnotami 0 a 1 (nie je / je daný pixel súčasťou regiónu) a vypočítaním priemernej hodnoty pre každý výstupný pixel. Výsledná matica sa môže previesť na obrázok v odtieňoch sivej vynásobením každého výsledného pixelu hodnotou 255. Tento obrázok je ešte vhodné invertovať aby oblasti obsahujúce regióny boli čierne. Takúto heatmapu je vidieť na obrázku 4.10.

¹Zdroj: <https://courses.lumenlearning.com/ws-u-sandbox/chapter/gestalt-principles-of-perception/>

²Zdroj: <http://www.fit.vutbr.cz/~ihynek/dashboards/doku.php?id=iis:2016>



Obr. 4.10: Ukážka heatmapy.

Bolo zistené, že ľudia vnímajú objekty značne podobným spôsobom (Gestalt princípy), ale sú viditeľné aj rozdiely spôsobené subjektívnym vnímaním, veľkosťou krátkodobej pamäti a podobne.

Kapitola 5

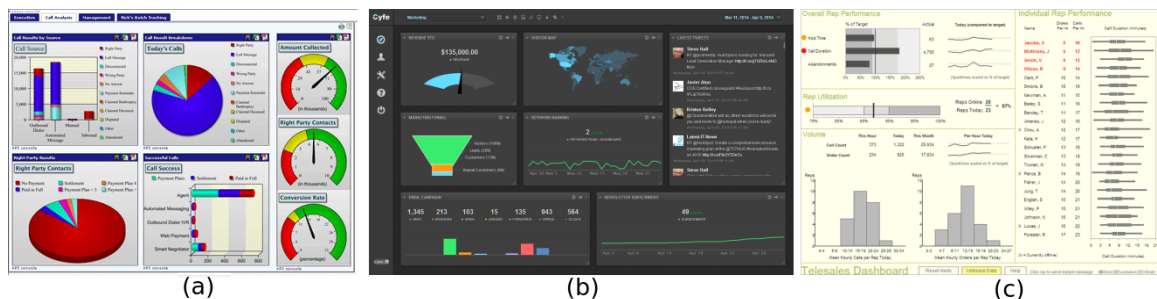
Metódy pre automatickú segmentáciu dokumentov

V tejto kapitole sa budeme zaoberať so segmentáciou dokumentov a ukážeme si niektoré konkrétne algoritmy segmentácie.

Segmentácia dokumentov je definovaná ako metóda na rozdelenie dokumentu na textové regióny a netextové regióny. [14] Takéto netextové regióny môžu obsahovať obrázky, grafy, tabuľky, atď. Segmentácia dokumentov má veľkú rolu v analýze dokumentov, nakoľko každý deň sa spracuje obrovské množstvo dokumentov. Dobrým príkladom je digitalizácia starých dokumentov a kníh pre archiváciu a ľahšie vyhľadávanie. Automatizovanie analýzy by ušetrilo veľa času, práce a peňazí. Samotná analýza sa skladá z troch krokov: segmentácia dokumentov na regióny, identifikácia typu regiónu a spracovanie jednotlivých regiónov zvlášť. Z týchto krokov sa zameriame na segmentáciu dokumentov. Existujú dva hlavné prístupy segmentácie dokumentov, metódy „zhora-nadol“ a „zdola-nahor“.

Metódy „zhora-nadol“ sú veľmi elegantné, rýchle a efektívne, ale dobré výsledky s týmito metódami je možné získať len pri dokumentoch s dobre známou štruktúrou. Začínajú spracovanie s celým dokumentom a snažia sa iteračne rozdeliť dokument do stĺpcov a riadkov, na základe geometrických informácií. [9] Metódy „zdola-nahor“ sú časovo náročnejšie a menej elegantné. Pracujú na úrovni pixelov bez toho, aby mali informácie o celom dokumente. Iteračne analyzujú dokument pixel po pixely a spájajú čierne pixely do spojených oblastí, ďalej do slov, potom do textových riadkov a nakoniec do textových blokov. [9]

Metódy segmentácie dokumentov väčšinou nevyžadujú komplikované predspracovanie alebo úpravy pred rozdelením dokumentu na regióny. Pri naskenovaných dokumentoch sa môže jednať o odstránenie šumu alebo zaostrenie obrazu. Vlastnosti, ktoré jednoznačne uľahčujú segmentáciu dokumentov sú: predpoklad bieleho pozadia, čierne písmo, známa štruktúra dokumentov (paragrafy, odrážky, titulky, nadpisy, atď). Oproti dokumentom u dashboardov je oveľa ťažšie nájsť vlastnosti, ktoré by umožnili segmentáciu, ako to vidieť na obrázku 5.1.



Obr. 5.1: Odlíšnosti dashboardov (a)¹(b)²(c) [5].

U dashboardov neexistuje jednotná štruktúra a limitácie grafických elementov. Používajú sa rôzne farby, odtiene, a pri spracovaní nastáva problém jednoznačne určiť farbu pozadia, ktoré oddeľuje jednotlivé regióny. Segmentáciu uľahčí, ak sa pozadie dá jednoznačne oddeliť od ostatných elementov. V dôsledku rôznorodosti dashboardov sa môže stať, že pozadie splynie s grafickými elementami. V takomto prípade je potrebné sa rozhodnúť, aké kroky sa použijú pri predspracovaní. Môže nastať prípad, keď výstup predspracovania má za následok správne segmentovaný dashboard, ale pri použití rovnakých krokov predspracovania na inom dashboarde výsledky segmentácie budú nesprávne. Predspracovanie a segmentácia dashboardov sú podrobne rozpracované v kapitole 6. Na samotnú segmentáciu dashboardov je možné použiť mierne modifikované algoritmy segmentácie dokumentov, ako metóda X-Y cut a RLSA.

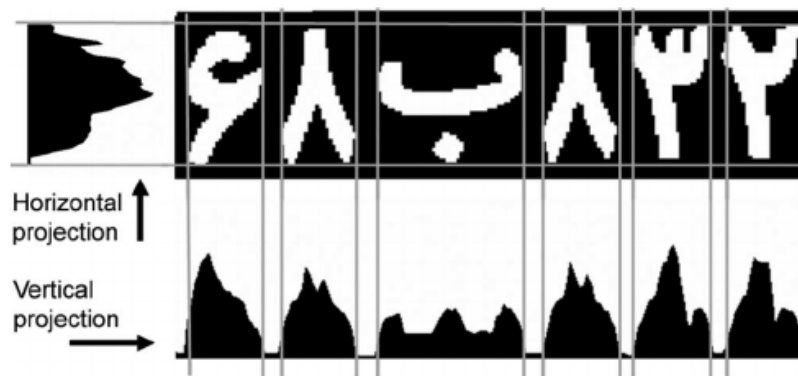
5.1 Metóda X-Y cut

Jedna metóda segmentácie „zhora-nadol“ známa ako rekurzívny X-Y cut rozkladá dokument postupne do množiny obdĺžnikových blokov. V každom kroku sú vypočítané profily pixelových projekcií v horizontálnom a vertikálnom smere. Obrázok 5.2 zobrazuje pixelovú projekciu. V jednoduchosti je možné definovať vertikálnu projekciu ako histogram počtu všetkých čiernych pixelov v stĺpcoch a horizontálnu projekciu ako histogram počtu všetkých čiernych pixelov v riadkoch. Potom sa rozdelenie zón vykonáva v najvýznamnejšom údolí vo vertikálnom či horizontálnom profile histogramu a proces je opakovaný až kým v oboch histogramoch je dostatočná možnosť rozdeliť ďalej obrazovku vertikálne respektíve horizontálne. Takéto možné vertikálne rozdelenia je možné vidieť na obrázku 5.2. Na základe tohto je možné popísať algoritmus X-Y cut v nasledujúcich krokoch:

1. Predspracovanie daného obrazu (naskenovaného dokumentu). (napr. odstránenie šumu, korekcia skreslenia)
2. Výpočet vertikálnej a horizontálnej pixelovej projekcie celého dokumentu.
3. Podľa vertikálnej alebo horizontálnej pixelovej projekcie sa obdĺžnik rozreže na dve časti vertikálne alebo horizontálne.
4. Pre nové obdĺžniky sú vypočítané nové pixelové projekcie. Ak je možné niektorý obdĺžnik ďalej „rezať“, tak sa pokračuje od kroku 3. V opačnom prípade nie je možné obrázok ďalej rozdeliť a aktuálny stav je finálny.

²Zdroj:<http://www.dashboards-for-business.com/dashboards-templates/business-intelligence>

²Zdroj:<https://www.semrush.com/blog/top-6-marketing-analytics-dashboard-tools/>

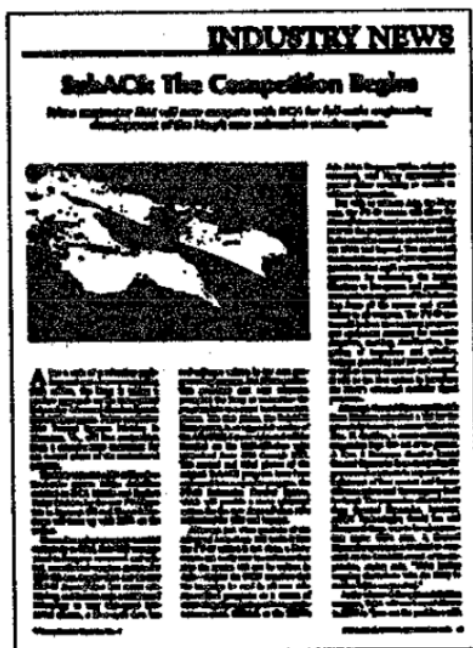


Obr. 5.2: Profily pixelových projekcií v horizontálnom a vertikálnom smere. [2]

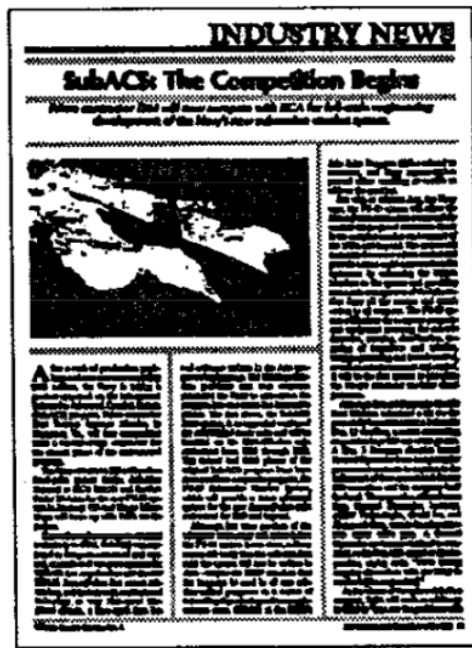
Štruktúrované rozdelenie obrazu dokumentu rekurzívnym X-Y cut umožňuje reprezentovať tento dokument vo forme stromu so vnorenými obdĺžnikovými blokmi - nazývaný *X-Y strom*. *X-Y strom* je štruktúra priestorových údajov, ktorá zobrazuje hierarchické rozdelenie stránky dokumentu rekurzívnym X-Y cut (X-horizontálne a Y-vertikálne). Koreňový uzol stromu X-Y je celá strana, obrázok. Každý uzol v strome predstavuje obdĺžnik na stránke. Deti uzla sú získané rozdelením obdĺžnika nadradeného uzla buď horizontálne alebo vertikálne s X-Y cut, ktoré sa striedavo používajú v nasledujúcich úrovniach stromu. Prvý pododdiel môže byť ľubovoľne nastavený horizontálne alebo vertikálne. Postup tvorby *X-Y stromu* je možné vidieť na obrázku 5.3.

V roku 1984 George Nagy a Sharad Seth navrhli *X-Y strom* ako možnosť reprezentácie rozloženia stránky [10]. Ich cieľom bolo vyvinúť schému na reprezentáciu rastrových, digitalizovaných (naskenovaných) dokumentov takým spôsobom, aby bola zachovaná nielen priestorová štruktúra vytlačeného dokumentu, ale malo by tiež uľahčiť automatické označovanie rôznych komponentov, ako sú text, čísla, titulky a titulky obrázkov a umožniť extrakciu dôležitých vzťahov (napr. poradie čítania komponentov) medzi nimi. Možné využitia sú: kompresia dát na prenos dokumentov a archiváciu, zadávanie dokumentov bez zopakovania do systémov na úpravu, formátovanie a vyhľadávanie informácií. *X-Y strom* má nasledujúce vlastnosti:

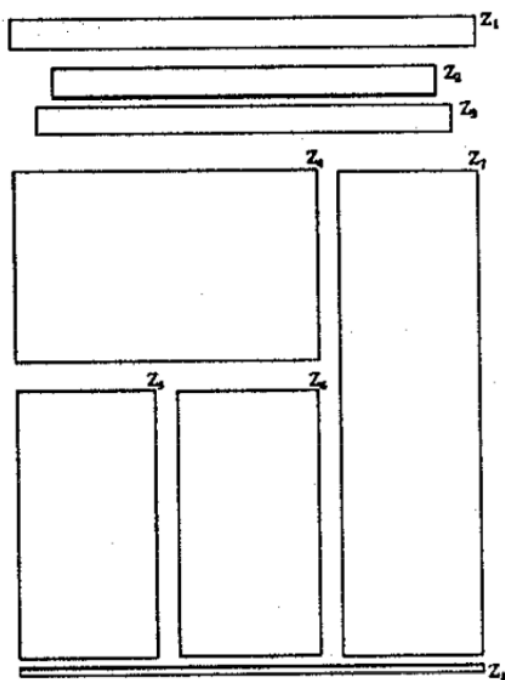
1. Je garantované, že stránka bude úplne rozdelená na obdĺžniky/kachličky, pričom nevynechá žiadnu časť. Vnorené subdivízie sa javia ako vhodné pre hierarchické štruktúry technických a obchodných záznamov.
2. Vytvárajú sa iba obdĺžniky, čo umožňuje rovnaké kroky spracovania na každej úrovni.
3. Na každej úrovni sa musí zväziť len lineárne (tj. buď horizontálne alebo vertikálne) delenie, čo umožňuje dobre riadené sekvenčné spracovanie blokov.
4. Na rozdiel od stromovej štruktúry *quadtree* [15] metóda nezávisí od výberu parametru a je viac citlivá na kontext
5. *X-Y strom* má tendenciu byť plytkejší než stromové štruktúry k-d, kde sú povolené iba binárne oddiely. Okrem toho binárne oddiely zvyčajne nezodpovedajú zmysluplným oddielom vytlačenej stránky.
6. *X-Y stromy* možno ľahko rozšíriť na tri alebo viac rozmerov; môže to byť výhodné pri riešení viacstránkových dokumentov a prípadne na iné použitie.



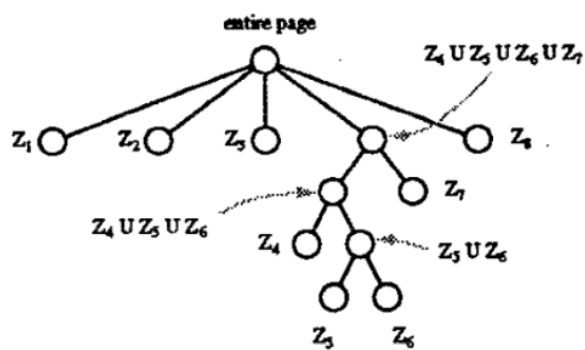
(a)



(b)



(c)



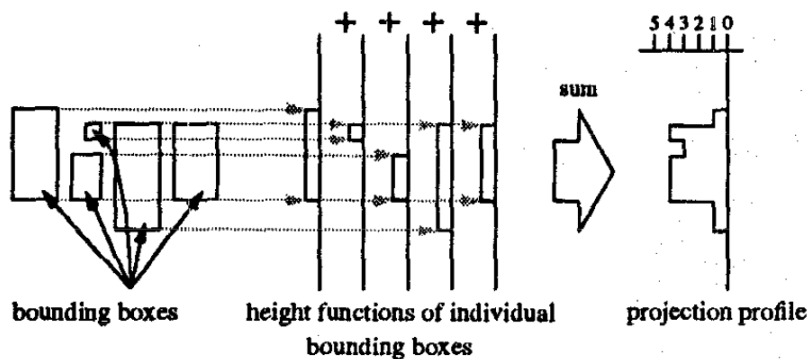
(d)

Obr. 5.3: Postup tvorby X-Y stromu.[6]

Nasledujúce podsekcie obsahujú rôzne spôsoby vylepšenia niektorých vlastností X-Y cut a tiež sa uvádza metóda využívajúca X-Y cut pre eliminovanie nedostatkov segmentácie podľa obrýsu.

5.1.1 Vylepšenie s použitím ohraničujúcich rámciekov

Toto vylepšenie navrhuje použitie ohraničujúcich rámciek spojených komponentov z čiernych pixelov namiesto počítania vertikálnej a horizontálnej projekcie zo samotných pixelov obrázku[6]. Veľká výhoda oproti pôvodnej metóde je dosiahnutá vo výpočte, nakoľko sa nepočíta s veľkým množstvom pixelov ale iba s ohraničujúcimi rámcami 5.4. Pochádza z roku 1995, keď počítače neboli natoľko výkonné ako dnes. Ďalší veľký rozdiel je aj možnosť využitia grafickej karty na paralelizáciu obecných výpočtov.



Obr. 5.4: Horizontálny profil projekcie, ktorý sa získa nahromadením ohraničujúcich boxov na zvislú čiaru. [6]

1. Predspracovanie daného obrazu, naskenovaného dokumentu (napr. odstránenie šumu, korekcia skreslenia).
2. Použitie algoritmu na označenie pripojených komponentov.
3. Obdržanie ohraničujúcich rámciek pre pripojených komponentov.
4. Vytvorenie koreňového uzla.
5. Výpočet horizontálnej a vertikálnej projekcie profilov.
6. Rozdelenie veľkých medzier v projekčných profiloch, ktorých šírka presahuje určitú prahovú hodnotu. Pri každom rozdelení sa vytvorí nový potomkový uzol. Pri každej úrovni rekurzívne sa striedajú horizontálne a vertikálne delenia.
7. Opakovanie krokov 5-6 rekurzívne, kým sú možné ďalšie delenia.

5.1.2 Vylepšenie segmentácie podľa obrysu využitím X-Y cut

1. Segmentácia podľa obrysu [7]

Ako iné segmentačné metódy, aj táto metóda sa snaží zrýchliť proces rozpoznania blokov záujmu, a to čím presnejšie. Samotná metóda pracuje s binárnym obrazom (čierny text, biele pozadie) na úrovni pixelov. Sila tejto fragmentácie pochádza z limitovaného priestoru spracovania, nakoľko vyhľadáva iba obrys paragrafov prechodom len krajných pixelov. K tomu využíva „okno“ fixnej veľkosti, čo môže spôsobiť dva problémy, keď je okno príliš malé alebo príliš veľké. V prípade, keď okno je malé, miesto paragrafov sa detekujú menšie jednotky ako slová alebo písmená. Keď je okno veľké tak sa spájajú väčšie jednotky ako aj celé paragrafy.

2. Identifikácia chybné segmentovaných bodov

Nakoľko segmentácia podľa obrysu je vykonaná v smere hodinových ručičiek, potenciálne chybné segmentované body sa nachádzajú v častiach, kde smer segmentácie je proti smeru hodinových ručičiek.

3. Využitie X-Y cut

Po identifikácii chybné segmentovaných bodov sa aplikuje na tieto časti metóda X-Y cut dvomi spôsobmi, a to vertikálne alebo horizontálne. Ak šírka prázdneho miesta medzi stĺpcami je väčšia ako šírka okna, tak sa začína horizontálnym delením. Naopak, ak sa táto šírka rovná šírke okna alebo je menšia, tak sa použije ako prvé vertikálne delenie.

5.2 Zdola nahor (Metóda RLSA)

V tejto sekcii si popíšeme príklad algoritmu segmentácie „zdola-nahor“, ktorá bola navrhnutá v článku [8]. Táto metóda sa skladá z nasledujúcich troch krokov:

1. Spojenie čiernych pixelov do blokov počas predspracovania.
2. Oddelenie textu od grafických elementov a označenie textových blokov.
3. Zlúčenie riadkov textu do paragrafov.

Najčastejšie používaná metóda na predspracovanie obrazu je RLSA (Run Length Smoothing Algorithm). Tento algoritmus spája susedné čierne oblasti, ktoré sú oddelené menej ako daným počtom pixelov (čo predstavuje prahovú hodnotu). Tento algoritmus sa aplikuje na obrázky v horizontálnom a vo vertikálnom smere. Výsledkom je prienik (logický AND) týchto dvoch obrázkov. Je to výpočtovo náročná metóda, a preto predspracovanie sa predĺži. Tiež je potrebné veľké množstvo pamäte na ukladanie týchto obrázkov. Ďalším problémom je výber hodnoty prahu. Je možné si priblížiť ideálnu hodnotu analyzovaním niektorých vlastností dokumentov, ale to by ďalej predlžoval predspracovanie.

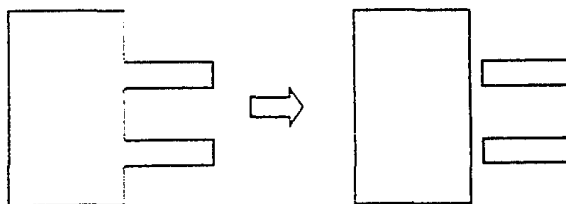
Navrhovaná metóda, založená na algoritmu RLSA, sa zaoberá týmito problémami a snaží sa podstatne zlepšiť rýchlosť a efektívnosť. Vykonáva sa iba horizontálna dilatácia a pôvodný obrázok sa zníži 8x horizontálne a 3x vertikálne (napr. obrázok s rozlíšením 2550x3300 sa zníži na 318x1100). Po redukcii možno uložiť celý obrázok do pamäte. Na urýchlenie analýzy sa vykonáva horizontálna dilatácia a redukcia obrazu súčasne.

Takto vytvorené bloky sú segmentované na text a grafické prvky podľa nasledujúcich pravidiel :

1. Bloky s výškou medzi $[H_{\min} .. H_{\max}]$ označ ako text, ostatné ako grafický element.
2. Rozdeľ bloky, ak rozdiel výšky riadkov je väčší ako 25%.
3. Odmietni blok, ak obsahuje v sebe iné bloky.
4. Bloky s hustotou medzi $[DS_{\min} .. DS_{\max}]$ označ ako riadok textu.

Prvé pravidlo odliší textový riadok od obrázku. Limity odpovedajú typickému použitiu veľkosti písma (väčšinou 8-20). Druhé pravidlo slúži na oddelenie blokov, ktoré boli chybné spojené počas predspracovania, ako vidieť na obrázku 5.5. Tretie pravidlo umožňuje získať text z obrázkov, bez čoho by bolo nemožné získať text z tabuliek alebo text obklopený

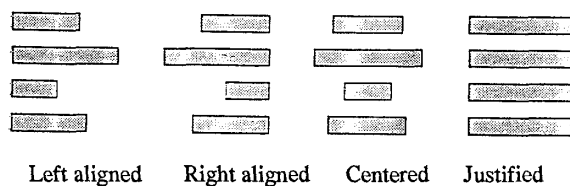
grafickými prvkami. Posledné pravidlo umožňuje rozdeliť bloky, u ktorých je podozrenie, že obsahujú text a bloky, ktoré obsahujú grafické prvky.



Obr. 5.5: Obrázok znázorňuje použitie druhého pravidla. [8]

Posledná časť tejto metódy spája detekované riadky do paragrafov, a to dvoma spôsobmi.

Prvý spôsob je založený na typografických vlastnostiach paragrafu, ako: typ zarovnania, veľkosť riadkovania, odsadenie a ďalšie parametre. Ďalej sa podľa „geografických“ pravidiel spájajú bloky, ktoré majú rovnaké zarovnanie, riadkovanie atď. a sú dostatočne blízko k sebe. Druhý spôsob používa informácie o logickej štruktúre dokumentov a pomáha identifikovať citácie, poznámky a podparagrafy v paragrafoch. Aj keď táto technika je jednoduchá, vyžaduje si zložité algoritmy. Existuje niekoľko odlišných zarovnaní textu, ale niektoré z nich sa používajú častejšie než iné. Základné 4 typy zarovnania textu sú zobrazené na obrázku 5.6.



Obr. 5.6: Obrázok znázorňuje zarovnanie paragrafov. [8]

Kapitola 6

Návrh metódy pre automatickú dekompozíciu dashboardov

Hlavným cieľom tejto práce je detekovať v dashboarde regióny reprezentujúce obálky vizuálne dominantných objektov. Tieto regióny sú použité k prevedeniu analýzy daného dashboardu. Postup celkovej analýzy je zobrazený na obrázku 6.1.

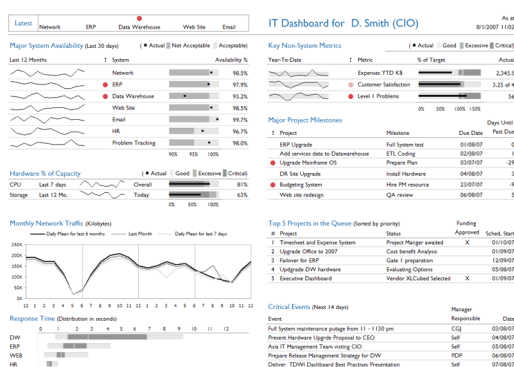
V tejto kapitole je popísaný návrh metódy automatickej dekompozície, nástroj na vyhodnotenie dashboardov, do ktorého bude integrovaná táto metóda a testovanie a overenie výsledkov.



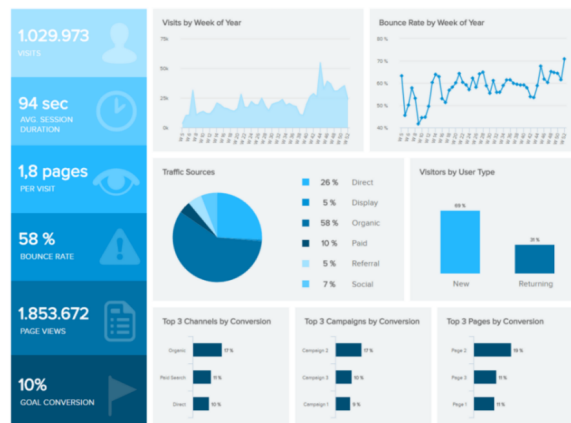
Obr. 6.1: Postup analýzy dashboardu.

6.1 Návrh metódy segmentácie dashboardov

V tejto práci som sa zamerlal na typy dashboardov, pri ktorých nie je možné využiť Gestalt princíp uzavretia. Jedná sa o dashboardy s uniformným pozadím a prvkami, ktoré nie je možné pospájať podľa pozadia, detekovaných čiar alebo uzavretosti. Prevedením farebného obrázku na čierne biely a doplnením dotýkajúcich sa čiernych pixelov vznikne veľké množstvo čiernych obdĺžnikov. Tieto dashboardy je možné segmentovať podľa Gestalt princípu blízkosti. Porovnanie dvoch dashboardov, ktoré odpovedajú tomuto popisu je vidieť na obrázku 6.2, kde obrázok (a) reprezentuje problematický dashboard z pohľadu segmentácie a obrázok (b) obsahuje regióny označené explicitne.



(a)



(b)

Obr. 6.2: Porovnanie dashboardov segmentovateľných podľa Gestalt princípu blízkosti (a) [5] a podľa Gestalt princípu uzavretia (b)².

Metriky rozvrhnutia pracujú s obálkami/objektami (regióny rozdelenej obrazovky) a majú ako výstup číselnú hodnotu, ktorá určí správnosť podľa danej metriky. Z toho plynie, že samotná segmentácia dashboardov je najkritickejšia časť celej analýzy, nakoľko zle detekované regióny majú za následok celkom odlišný výsledok evaluácie.

Ako vidieť z príkladov dashboardov uvedených na obrázku 5.1, navzájom sa veľmi líšia, a to nie len farbou, použitými grafickými prvkami ale aj svojou štruktúrou. Z toho plynie, že pred samotnou segmentáciou je potrebné tieto dashboardy vhodným spôsobom upraviť. Existuje niekoľko možností úprav, preto je potrebné najprv analyzovať obrázok a na základe výsledkov analýzy vybrať najvhodnejšie kroky predspracovania.

Možné kroky predspracovania:

- prahovanie;
- prevod obrazu do stupňov sivej;
- redukcia počtu farieb (posterizácia);
- podľa histogramu farby je možné invertovanie farby. (Ideálne je získanie bieleho pozadia);
- úprava kontrastu pre lepšie výsledky detekcie hrán;
- odstránenie farebného prechodu a farebného pozadia.

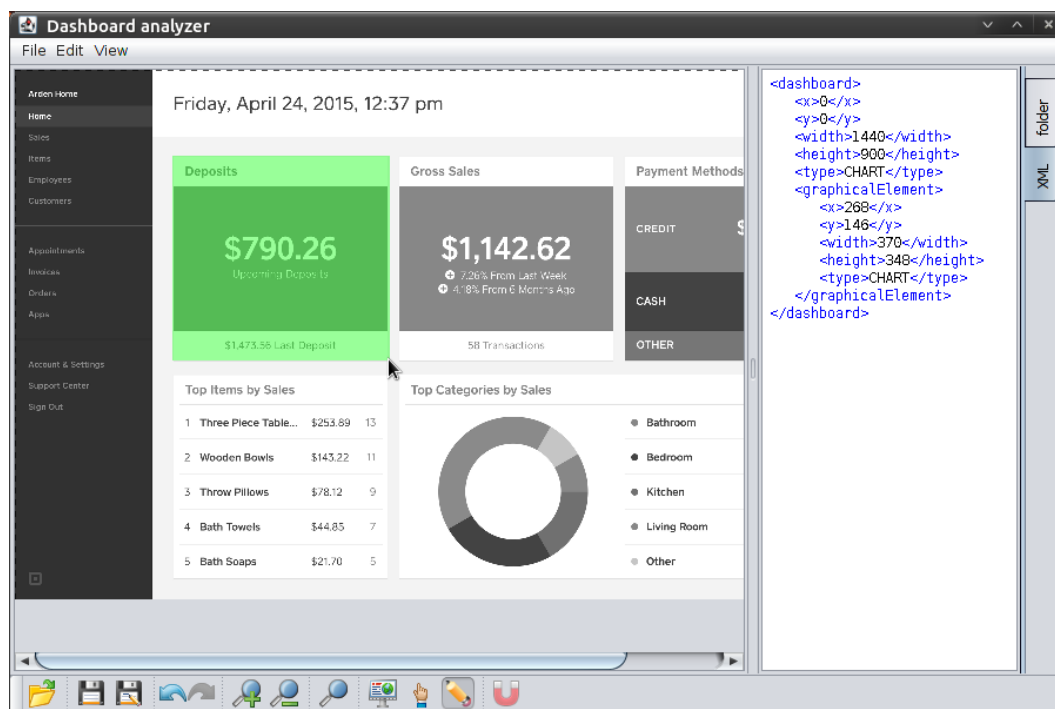
Najjednoduchšia metóda segmentácie obrazu je prahovanie [3]. Prahovanie je založené na hodnotení jasú každého pixelu. Základom je nájdenie takej hodnoty, tzv. prahu v histogramu, pre ktorú platí, že všetky hodnoty vyššie ako prah odpovedajú poprediu a všetky hodnoty jasú nižšie ako prah zodpovedajú pozadiu. V mnohých prípadoch je postačujúce urobiť prahovanie, ak nie tak je potreba ďalej analyzovať dashboard a vybrať ďalšie možnosti, prípadne ich kombinácie.

²Zdroj: <https://www.datapine.com/dashboard-examples-and-templates/marketing>

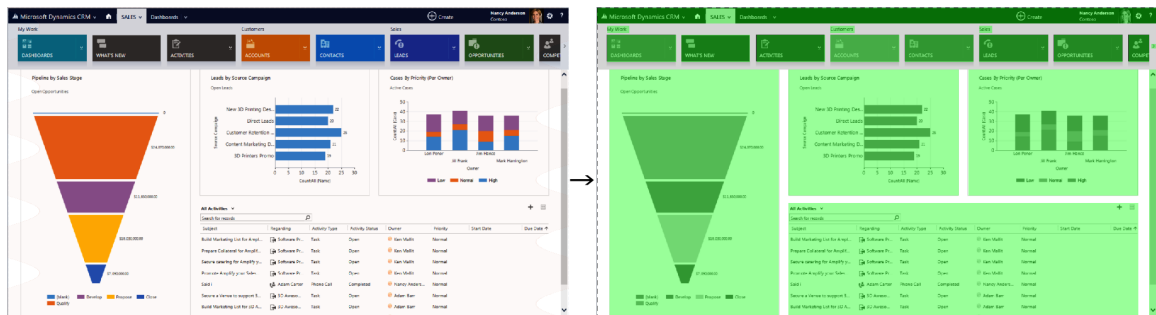
Po týchto úpravách nasleduje prevod na čierne biely obraz, na ktorý už je možné použiť algoritmus segmentácie, ktorý bude postavený nad prístupom „zdola nahor“ pre získanie regiónov záujmu.

6.2 Nástroj na vyhodnotenie dashboardov

Na vyhodnotenie dashboardov použijeme existujúci nástroj, ktorý je vidieť na obrázku 6.3. Tento nástroj poskytuje API, ktoré obsahuje funkcie pre načítanie, zobrazenie dashboardov, jednoduché funkcie ako zmena veľkosti a vrátenie zmeny. Ďalej obsahuje funkciu pre ručné označenie, zobrazenie a uloženie vo formáte XML regiónov, metriky vyhodnotenia týchto regiónov. Nástroj tiež obsahuje veľa ďalších užitočných funkcií ako vytváranie heatmap zo sady segmentovaných dashboardov a funkcie pre rôzne spracovania obrazu. Algoritmy a predspracovanie pre automatickú segmentáciu dashboardov budú implementované ako modul v tomto nástroji, s využitím poskytovaných funkcií tohto nástroja. Výstup segmentácie bude XML súbor popisujúci regióny. Tieto regióny zobrazuje nástroj ako je zobrazené na obrázku 6.4. Vyhodnotenie výsledkov sa bude konať na základe týchto XML popisov.



Obr. 6.3: Nástroj na vyhodnotenie Dashboard.



Obr. 6.4: Výsledok segmentácie Dashboard³.

6.3 Formát XML výsledku segmentácie

XML (eXtensible Markup Language) [1] je značkovacím jazykom odporúčaný W3C (World Wide Web Consortium), ktorý popisuje dáta vhodným spôsobom na uloženie a prenos. Má jednoduchú štruktúru, ktorá sa skladá zo značkovacích symbolov (podobá sa na HTML), ktoré popisujú vlastnosť a zároveň aj obsahujú dáta. Nazýva sa to „*self-describing/self-defining*“. Tieto značky môžu byť vnorené a ľubovoľne pomenované tak, aby vyhovovali popisu danej úlohy alebo popisu dát formou, na ktorej bolo dopredu dohodnuté. Umožňuje dynamické spracovanie dokumentu bez potreby dopredu pripraviť štruktúru dát. Je to popis vytvorený tak, aby bol čitateľný človekom aj počítačom.

Pre popis výsledných regiónov bol definovaný formát v XML, ktorý je možné rozdeliť na dve časti, popis dashboardu a popis jednotlivých regiónov.

- Následujúca ukážka zobrazuje definíciu dashboardu:

```
<dashboard>
  <x>0</x>
  <y>0</y>
  <width>1440</width>
  <height>900</height>
  <type>CHART</type>
  ...
</dashboard>
```

kde x a y reprezentujú koordináty x a y dashboardu (vždy $[0,0]$), $width$ a $height$ reprezentuje šírku a výšku dashboardu. Namiesto riadku „...“ môže nasledovať ľubovoľné množstvo popisov regiónov.

³Zdroj:<http://www.fit.vutbr.cz/~ihynek/dashboards/doku.php?id=iis:2016>

- Následujúca ukážka zobrazuje definíciu regiónov:

```

...
<graphicalElement>
  <x>268</x>
  <y>146</y>
  <width>370</width>
  <height>348</height>
  <type>CHART</type>
</graphicalElement>
...

```

kde x a y reprezentujú koordináty x a y regiónu, $width$ a $height$ reprezentujú šírku a výšku regiónov.

6.4 Návrh testovania a overenia výsledkov

- **Návrh testovania**

Jednou z úloh pri tvorbe metódy automatickej segmentácie je zostaviť sadu dashboard takého typu ako je popísané v sekcii 6.1, vhodných na testovanie užívateľmi. Pri testovaní sa očakáva od užívateľov, aby označili vizuálne dominantné regióny. Z týchto výsledkov sa vytvoria obrázky znázorňujúce „priemerné“ označenie regiónov (ďalej *heatmapa*), ktoré budú slúžiť jednak ako výsledky, ktoré chceme dosiahnuť pri automatickej segmentácii (referenčné riešenie) a tiež sú vhodné na overenie výsledkov. Aplikácia ponúka tri režimy zobrazenia obrázku.

1. Režim ukážky základného farebného obrázku 🖼️.
2. Režim ukážky regiónov s možnosťou vymazať, zmeniť veľkosť a pozíciu regiónu 🖱️.
3. Režim ukážky regiónov s možnosťou vymazať a pridať nové regióny ✏️.

Metóda automatickej segmentácie bude pridaná pod *Segmentation* → *BottomUp* alebo ikonou 📊.

Prvou úlohou užívateľa je oboznámiť sa s frameworkom. Postup pri testovaní užívateľom je nasledovný:

1. Otvoriť zložky s dashboardami.
2. Vybrať dashboard zo zoznamu na pravej strane.
3. Prepnúť aplikáciu do režimu kreslenia.
4. Označiť regióny.
5. V prípade potreby editovať regióny v režime editácia.
6. Uložiť označené regióny.

Pri tvorbe frameworku pre analýzu dashboardov podobný výskum už bol prevedený, ktorý je popísaný v sekcii 4.4. Užívatelia mali k dispozícii sadu dashboard, na ktorých mali za úlohu označiť vizuálne dominantné regióny. Mal som k dispozícii výsledné obrázky so XML popisom reprezentujúce regióny vyznačené užívateľmi.

Ako bolo zmienené v kapitole 4, každý človek je unikátny s rozličnou krátkodobou pamäťou. Ľudia vnímajú svoje okolie trochu iným spôsobom, odlišuje sa ako zhlukujú menšie celky do objektov. Tento jav sa ukázal na výsledkoch od užívateľov, ktoré sa mierne líšia.

Ďalej sa bude určovať entropia medzi vzorkami od užívateľov. Entropia je definovaná ako miera neurčitosti. To znamená, že pri porovnaní vzoriek od užívateľov vysoká entropia sa vyskytne v miestach kde sa tieto vzorky najviac líšia.

- **Overenie výsledkov**

Overenie výsledkov sa uskutoční porovnaním výsledku automatickej dekompozície s veľkým množstvom vzoriek získaných z výskumu. Výsledky dekompozície sa budú vyhodnocovať dvoma spôsobmi.

Prvý spôsob:

1. Porovnaním výslednej matice, získanej automatickou segmentáciou a heatmapy od užívateľov získame odchýlku.
2. Výpočet binárnej entropie medzi vzorkami od užívateľov.
3. Cieľom je získať nižšiu odchýlku ako je entropia medzi vzorkami od užívateľov.

Druhý spôsob:

1. Výpočet metrík, ktoré vyhodnocujú rozloženie užívateľského rozhrania na základe výsledkov segmentácie.
2. Výpočet priemernej hodnoty metrík zo získaných vzorov od užívateľov.
3. Porovnanie výsledkov bodu 1 a bodu 2.

Kapitola 7

Implementácia

V tejto kapitole uvádzam použité nástroje, podrobnejšie popisujem metódu automatickej dekompozície a kroky predspracovania.

7.1 Použité nástroje

Pre analýzu dashboardov bol využitý existujúci framework, ktorý je implementovaný v objektovo orientovanom programovacom jazyku JAVA. Metóda dekompozície obrazovky z dôvodu jednoduchšej integrácie bola tiež implementovaná v jazyku JAVA. Rozširuje triedu *AbstractSegmentationAlgorithm* a implementuje triedu *ISegmentationAlgorithm*. Táto trieda je integrovaná do frameworku pomocou konfiguračného súboru.

Pre implementáciu dekompozície bol vybraný postup zdola-nahor, ktorý je pomalší, ale jeho výsledky sa ukázali byť lepšie ako implementácia založená na algoritmu XY-cut.

7.2 Metódy automatickej dekompozície a predspracovanie

V tejto sekcii opisujem použité kroky predspracovania, segmentáciu obrazu a dôležité funkcie využívané pri segmentácii.

7.2.1 Predspracovanie

Použité kroky predspracovania:

- **Odstránenie farebných prechodov** – Táto funkcia sa používa v prípadoch, keď pozadie dashboardu obsahuje rôzne farby a prechody medzi týmito farbami. Slúži na odstránenie farebného pozadia dashboardu. Môže nastať prípad, keď obrázok obsahuje pixely pozadia s odlišnou hodnotou, ktoré ale sú neviditeľné voľným okom a úplne zapadnú do pozadia. Pri spracovaní obrazu sa môžu tieto pixely prejaviť neskôr, keď sú nesprávne rozpoznané. V metóde automatickej dekompozície, ktorá je zameraná na konkrétny typ dashboardov 6.1 sa táto funkcia používa na odfiltrovanie týchto pixelov.
- **Prevedenie do stupne sivej** – Táto funkcia sa využíva na redukovanie farieb. Týmto uľahčí spracovanie obrazu, nakoľko sa ďalej pracuje s obrázkom, ktorý má hodnoty od 0 do 255. Jedná sa o značne menší počet hodnôt ako v pôvodnom formáte RGB, ktorý obsahuje 3 kanály ($3 * 8\text{bitov}$, 2^{24} farieb).

- **Posterizácia** – Táto funkcia sa používa na redukovanie hodnôt farieb reprezentovaním niekoľkých odtieňov originálneho obrazu ako jedna farba. Pri segmentácii sa využíva napodobňovanie spôsobu, ako človek zhlukuje objekty, pričom malé rozdiely v odtieňoch farby si nevšima. Na druhej strane ale počítač tieto hodnoty berie striktné a môže nastať prípad, že sa rozdelia objekty ktoré by človek vnímal ako jeden celok.
- **Prahovanie** – Táto funkcia sa využíva na prevedenie obrazu na čierne biely. Základ tejto metódy už bol popísaný v sekcii 6.1. Pri segmentácii sa využíva modifikácia tejto funkcie, keď sa ako prah berie najčastejšia hodnota z histogramu, ktorá s veľkou pravdepodobnosťou predstavuje pozadie.

7.2.2 Hlavná funkcia

Vstupom tejto funkcie je dashboard vo formáte `BufferedImage`, ktorý je prevedený na maticu s hodnotami `Integer`. Výsledok segmentácie je získaný kombináciou matíc dvoch metód `bottomUp1` a `bottomUp2` a následným spracovaním tohto výsledku funkciou `connectSmallRegions`.

- Prvá metóda, `bottomUp1` hľadá podobnosť medzi rozpoznanými obdĺžnikmi podľa polohy a veľkosti. Začína predspracovaním vstupného obrazu odstránením farebného prechodu, prevedením do stupne sivej, posterizáciou a prahovaním prevedie na čierne biely obrázok. Z čiernych pixelov sú vytvorené obdĺžniky. Najprv sú vyhľadane krajné body v súvislých čiernych tvaroch. Podľa týchto bodov sú pixely prekryté obdĺžnikom, prepísaním na čiernu farbu. Z týchto obdĺžnikov sú vytvorené zoznamy, ktoré môžu byť potenciálne spojené do riadkov, stĺpcov. Tieto zoznamy sa zvlášť analyzujú. Pre každý zoznam sa vypočíta limit vzdialenosti medzi obdĺžnikmi, podľa ktorého sú pospájané. Tento postup vidieť vo forme pseudo kódu 1.

Algorithm 1 BottomUp1

Input: `BufferedImage` of Dashboard(RGB)

Output: Black and white matrix with regions

```

convert BufferedImage to int matrix
// Preprocess
remove gradient from image
convert image to grayscale
posterize image
convert image to black and white // threshold using most frequent value

create rectangles from black pixels // also removes points and small lines
join close rectangles in rows
redraw rectangles
join close rectangles in columns
redraw rectangles

```

- Druhá metóda `bottomUp2` používa odlišné predspracovanie, ktoré neodstráni toľko detailov ako prvá metóda. Z tejto bitmapy sa vypočíta vertikálny a horizontálny limit blízkosti medzi susednými čiernymi pixelmi. Všetky obdĺžniky, ktoré sa nachádzajú

v blízkosti pod týmto limitom sú spojené. Tento postup vidieť vo forme pseudo kódu [2](#).

Algorithm 2 BottomUp2

Input: BufferedImage of Dashboard(RGB)

Output: Black and white matrix with regions

```
convert BufferedImage to int matrix
// Preprocess
remove gradient from image
convert image to black and white // threshold using most frequent value

create rectangles from black pixels // also removes points and small lines
calculate horizontal and vertical threshold
join rectangles with smaller closeness than threshold
```

- Výsledná sada rozpoznaných regiónov sa vytvorí kombináciou výsledkov z prvej a druhej funkcie. Nakoniec sa analyzujú regióny podľa veľkosti so snahou ďalej zhlukovať ich. Tento postup vidieť vo forme pseudo kódu [3](#) a na obrázkoch medzikrokov na obrázku [7.1](#).

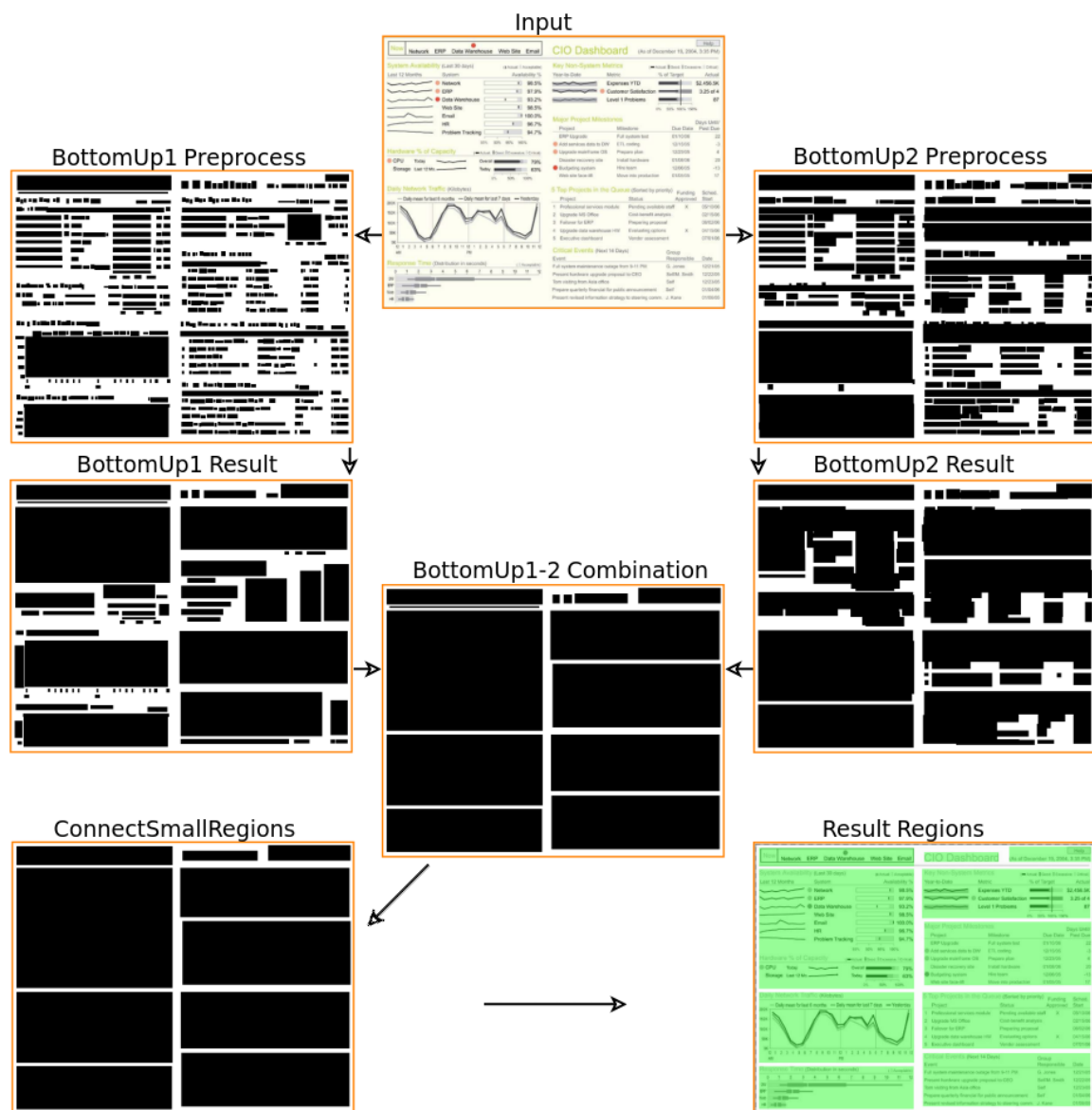
Algorithm 3 MainFunction

Input: BufferedImage of Dashboard(RGB)

Output: List of final regions

```
initialize class variables according to input
get resultMatrix1 from bottomUp1
get resultMatrix2 from bottomUp2
combine resultMatrix1 and resultMatrix2

create rectangles from black pixels // also removes points and small lines
repeat
    connect remaining small regions using threshold
until current result != previous result
redraw rectangles
create list of regions
```



Obr. 7.1: Vizualizácia postupu metódy automatickej dekompozície obrázkami medzivýsledkov.

7.2.3 Utility použité pri segmentácii

Pri implementácii boli využité rôzne utility. Nakoľko RGB obrázkov je prevedený na maticu s hodnotami [0,255] (čierno biele), väčšina utilít obsahuje operácie nad maticami. Niektoré sú súčasťou frameworku a nasledujúce boli implementované v triede `BottomUpUtil`:

- **setArray** – Funkcia prepíše hodnoty matice na zadanú hodnotu (použitie napr. na prekreslenie obrázku na biele).
- **averageMatrix** – Funkcia vráti priemer čiernych pixelov v obrázku.
- **differentMatrices** – Funkcia porovná matice či sú odlišné.
- **sortDoubleList** – Funkcia usporiada dvojité zoznam¹.
- **createRectangles** – Funkcia prebraná z triedy `GrayMatrix` a upravený limit maximálnej veľkosti obdĺžnikov.

Ďalej si rozpišem niektoré zložitejšie funkcie.

- **getRegions** – Funkcia vráti zoznam regiónov reprezentovaných triedou `Region`, ktoré berie zo vstupného obrazu. Vyhľadáva najväčšie čierne oblasti tvaru obdĺžnika. Trieda `Region` obsahuje `x` a `y` koordináty ľavého horného rohu, šírku a výšku obdĺžnika a rozširuje triedu `Rectangle` funkciou, ktorá vráti obsah regiónu. Regióny s menšou šírkou alebo výškou ako 3 sú brané ako potenciálne čiary alebo výsledok nekvalitného obrazu a vynechávajú sa z výsledného zoznamu.
- **getLimit** – Funkcia vráti maximálnu horizontálnu alebo vertikálnu vzdialenosť medzi regiónmi, ktoré boli označené ako potenciálne patriace do jedného riadku alebo stĺpcu. Hodnota je stanovená ako polovica priemernej vzdialeností medzi regiónmi.
- **getTreshold** – Funkcia vráti horizontálny a vertikálny prah, ktorý je počítaný zo vzdialenosti medzi dvoma susednými čiernymi pixlami oddelených bielymi pixlami. Tento prah využívajú funkcie `BottomUp2` a `connectSmallRegions`.
- **getMinRegionSize** – Funkcia vráti polovičnú hodnotu priemeru veľkosti všetkých rozpoznaných regiónov. Regióny pod týmto limitom sú spojené s regiónmi, ktoré sa nachádzajú od nich na menšiu vzdialenosť ako hodnota prahu.
- **joinRows** – Funkcia pripraví dvojité zoznam inštancie triedy `Region`, ktoré sa potenciálne dajú spojiť do riadkov, limit blízkosti týchto regiónov a zavolá funkciu `reDrawRowsOrColumns` v horizontálnom smere.
- **joinColumns** – Funguje rovnako, ako funkcia `joinRows`, ale vo vertikálnom smere.
- **reDrawRowsOrColumns** – Táto funkcia v zadanom dvojitém zozname Regiónov pospája ich vertikálne alebo horizontálne, podľa zadaného smeru, ak sú dostatočne blízko seba podľa zadaného limitu.
- **connectSmallRegions** – Funkcia podľa vypočítaných horizontálnych a vertikálnych limitov a minimálnej veľkosti regiónov sa snaží pospájať regióny pod týmto limitom, ktoré doteraz neboli spojené.

¹List<List<Region>>

Kapitola 8

Testovanie a vyhodnotenie

Táto kapitola obsahuje v prvej časti stručný popis spôsobu testovania a v druhej časti vyhodnotenie výsledkov.

8.1 Testovanie

V rámci tejto diplomovej práce bola navrhnutá konkrétna metóda dekompozície užívateľského rozhrania. Testovanie nebolo zamerané na odskúšanie tejto metódy užívateľmi, ale získať čo najviac vzorov segmentácie dashboardov od užívateľov. K tomu bol použitý nástroj popísaný v sekcii 6.2. Nakoľko sa jedná o aplikáciu napísanej v multiplatformovom programovacom jazyku JAVA, testovanie prebehlo na rôznych operačných systémoch podľa výberu užívateľa (Linux, Windows, MAC). Jedinou podmienkou bola verzia Javy. Potrebné bolo mať nainštalovanú verziu 1.8 alebo vyššiu verziu. V rámci výskumu 4.4 prebehlo takéto testovanie študentmi. Nakoľko pri implementácii som sa zameral na konkrétny typ dashboardov popísaných v kapitole 6.1, z výslednej sady dashboardov som vybral tie, ktoré odpovedali tomuto popisu dashboard. Konečné vyhodnotenie sa uskutočnilo porovnaním výsledku automatickej dekompozície s veľkým množstvom vzoriek získaných z výskumu. Výsledky dekompozície sú vyhodnotené porovnaním regiónov a porovnaním výsledkov metrík.

Boli použité aj ďalšie dashboardy, webové stránky a dokumenty na vyskúšanie implementácie automatickej dekompozície, ktoré neboli vyhodnotené užívateľmi. Tieto vzorky boli použité na testovanie a trénovanie algoritmu pri implementácii. Príklady sú pridané ako prílohy C.1, C.2

8.2 Vyhodnotenie výsledkov

Z veľkého množstva dashboardov, ktoré boli použité pri výskume 4.4, bolo vybraných 11. Ich vlastnosti odpovedali popisu v sekcii 4.4. Sú pomenované podľa vzoru $aXXX$ a $bXXX$, kde XXX reprezentuje číslo dashboardu. Túto testovaciu množinu obsahuje okrem iného príloha A. Kvôli jednoduchosti si zavedieme výrazy $výsledky_{AD}$ (výsledky automatickej dekompozície) a $výsledky_U$ (výsledky od užívateľov), ktoré ďalej budeme používať v tejto sekcii.

8.2.1 Vyhodnotenie porovnaním regiónov

K pochopeniu výsledkov sú potrebné heatmapy 4.4 a entropia z výsledkov_U. Najprv je potrebné výsledky_U previesť na heatmapu, z ktorej sa počíta binárna entropia. Binárna entropia má najvyššiu hodnotu v bode, kde pravdepodobnosť výskytu 0 alebo 1 je rovnaká. Prevedenie výslednej matice prebehne podobným spôsobom ako u heatmapy. Príklad entropie v dashboardu je znázornený na obrázku 8.1



Obr. 8.1: Ukážka entropie z výsledkov_U

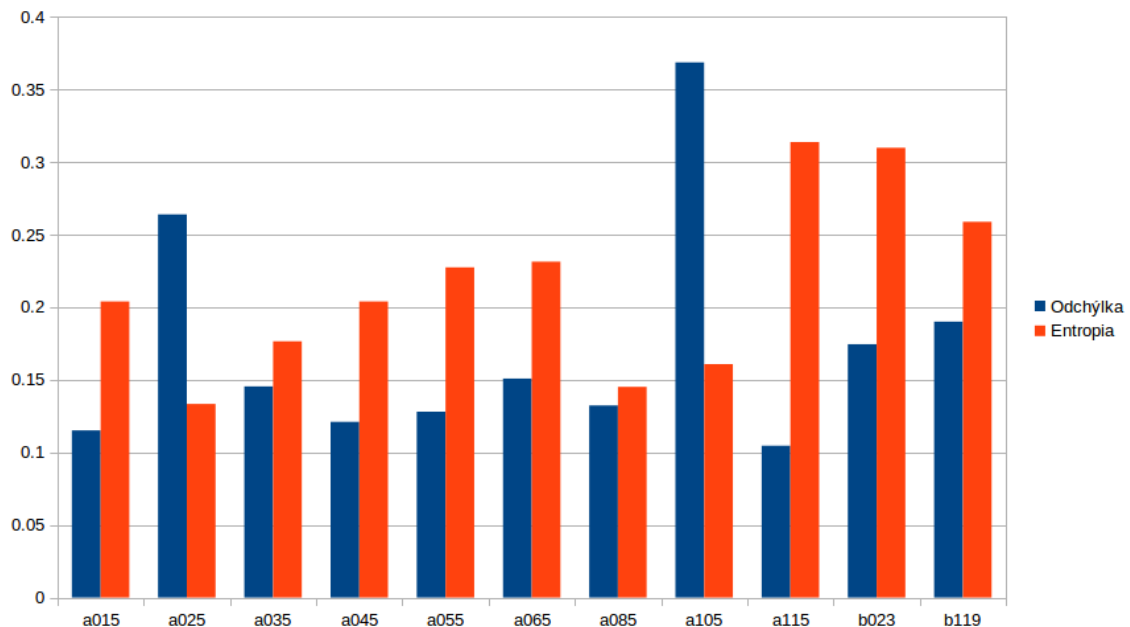
Odchýlku výsledkov_{AD} od výsledkov_U je možné získať dvomi spôsobmi:

1. Prevedenie výsledkov_{AD} a výsledkov_U na maticu s hodnotami 0 a 1. Porovná sa každá matica z výsledkov_U s maticou z výsledku_{AD}. Odchýlku tvorí priemer rozdielov všetkých matíc.
2. Prevedenie výsledkov_{AD} na maticu s hodnotami 0 a 1 a vytvorenie heatmap z výsledkov_U. Z výsledkov_{AD} sa odčíta heatmapa, ktorá je už matica s hodnotami pravdepodobnosti výskytu pixelu (0-1). Odchýlku tvorí priemer všetkých pixelov vo výslednej matici.

Pre overenie správnosti výsledkov sa porovná odchýlka (získaná ako bolo popísané vyššie) s entropiou získanou z výsledkov_U. Cieľom je aby odchýlka bola menšia ako entropia vo výsledkoch_U. Tieto výsledky sú uvedené v tabuľke 8.1 a v podobe grafu sú zobrazené na obrázku 8.2.

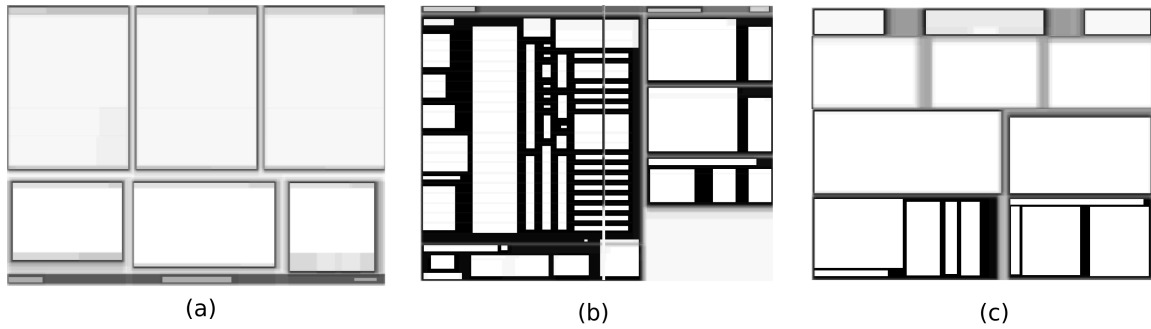
ID	Odchýlka	Entropia	Správnosť
a015	0.1151	0.2039	✓
a025	0.2639	0.1333	✗
a035	0.1454	0.1765	✓
a045	0.1210	0.2039	✓
a055	0.1280	0.2275	✓
a065	0.1508	0.2314	✓
a085	0.1323	0.1451	✓
a105	0.3686	0.1608	✗
a115	0.1046	0.3137	✓
b023	0.1745	0.3098	✓
b119	0.1900	0.2588	✓

Tabuľka 8.1: Tabuľka s hodnotami vyhodnotenia



Obr. 8.2: Graf porovnania odchýlky získanej z výsledkov_{AD} a entropiou výsledkov_U. (cieľ: odchýlka < entropia)

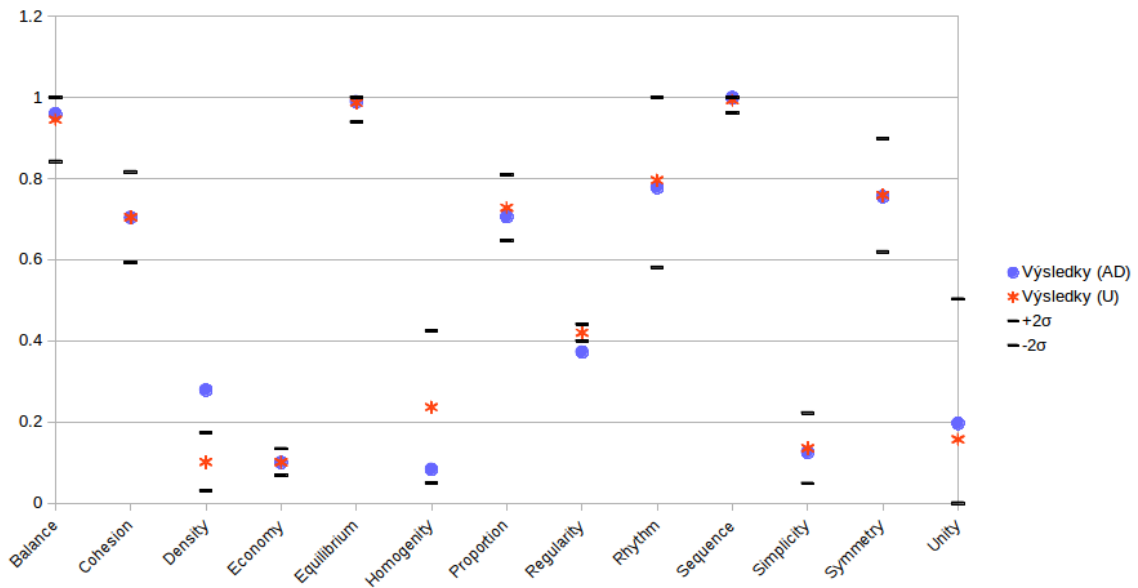
Čím je väčšia hodnota entropie od hodnoty odchýlky, tým je lepší výsledok dekompozície, respektíve čím je väčšia hodnota odchýlky od hodnoty entropie, tým sú výsledky horšie. Na obrázkoch 8.3 je vidieť porovnanie výsledných heatmap získaných porovnaním výsledkov_{AD} a heatmap od užívateľov, od najlepšieho výsledku k najhoršiemu. Cieľom je, aby entropia a odchýlka boli čo najmenšie a súčasne odchýlka menšia, ako entropia. V prípade, že by bola entropia rovná 1, nebolo by možné daný dashboard analyzovať týmto spôsobom, preto že sa užívatelia nezhodli na regiónoch.



Obr. 8.3: Porovnanie heatmap, keď: (a) odchýlka < entropia, (b) odchýlka \approx entropia (c) odchýlka > entropia.

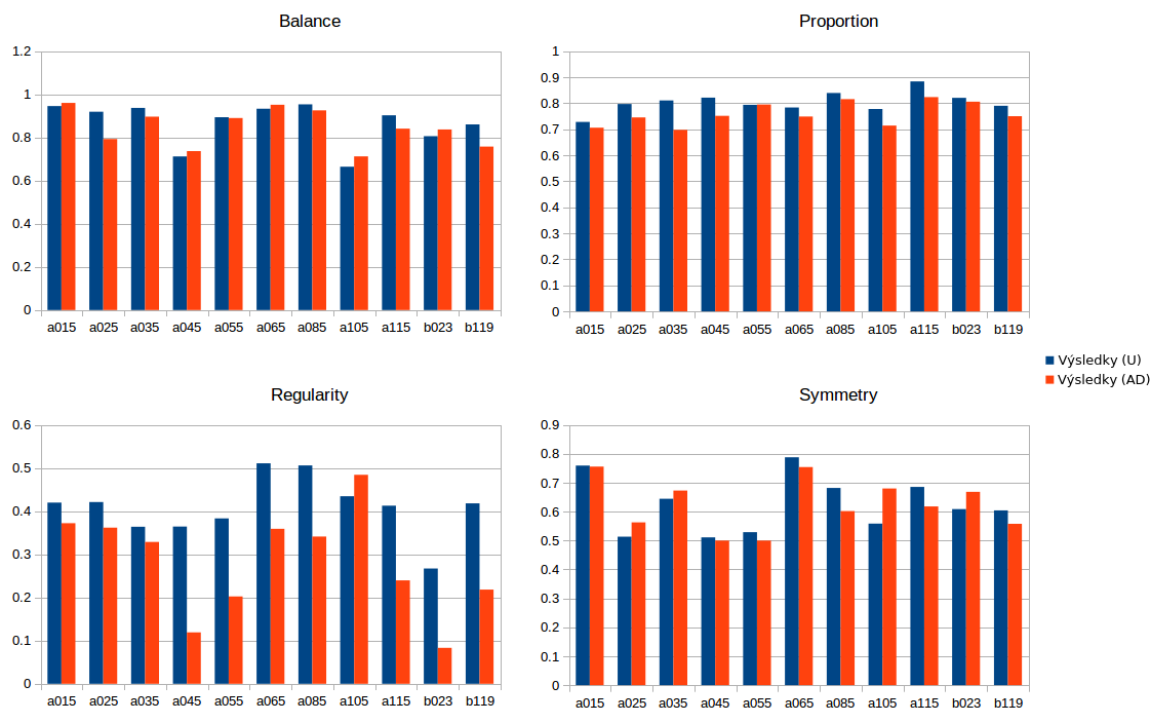
8.2.2 Vyhodnotenie porovnaním výsledkov metrík

Pre výsledky_{AD} a výsledky_U boli vypočítané metriky. Pre porovnanie sme vytvorili priemernú hodnotu zo všetkých výsledkov metrík pre daný dashboard od všetkých užívateľov. Ďalej bola vypočítaná smerodajná odchýlka z výsledkov metrík od užívateľov. Tieto hodnoty je možné vidieť vo forme grafu na obrázku 8.4, kde hodnoty $\pm 2\sigma$ sú limitované v rozsahu $[0,1]$, nakoľko aj výsledky metrík sú obsiahnuté v tomto rozsahu.



Obr. 8.4: Graf zobrazujúci priemernú hodnotu metrík od užívateľov, smerodajnú odchýlku a výsledky metrík automatickej segmentácie.

Z vyššie uvedených výsledkov vyplýva, že aj keď metriky boli počítané pre ten istý dashboard s rovnakými výsledkami segmentácie od užívateľov a automatickej segmentácie, výsledky pre niektorých metrík sa líšia jednoznačne viac ako u iných. Z toho je zrejmé, že metriky sú rôzne citlivé na menšie rozdiely v usporiadaní regiónov a ich veľkosti. Na záver je ukázaný obrázok 8.5 obsahujúci výsledky všetkých skúmaných dashboardov pre štyri rôzne metriky.



Obr. 8.5: Zobrazenie štyroch rôznych metrík s hodnotami z automatickej segmentácie a od užívateľov.

Na grafoch sú znázornené výsledky od užívateľov a z automatickej segmentácie podľa štyroch rôznych metrík. Z pohľadu vyváženosti, pomeru a symetrie výsledky sa značne menej líšia, ako z pohľadu pravidelnosti. Z toho vyplýva, že výsledky automatickej segmentácie lepšie vyhovujú vyhodnoteniu podľa metrík, ktoré sú menej citlivé na menšie rozdiely vo veľkosti rozlohy a polohy regiónov.

Kapitola 9

Záver

V tejto práci bola navrhnutá metóda automatickej dekompozície obrazovky na regióny reprezentujúce obálky vizuálne dominantných objektov. Podľa týchto regiónov je možné previesť analýzu užívateľského rozhrania metrikami rozvrhnutia.

V prvej časti je podrobne opísaný dashboard a vlastnosti dashboardov. Zaoberá sa vizuálnym vnímaním, metrikami rozvrhnutia a metódami automatickej segmentácie dokumentov.

Následne sa v druhej časti práce venujem návrhu, implementácii a vyhodnoteniu segmentačnej metódy, ktorá je zameraná na dashboardy s uniformným pozadím a prvkami, ktoré nie je možné pospájať podľa pozadia, detekovaných čiar alebo Gestalt princípu uzavretosti. Prevedením farebného obrázku na čierne biely a doplnením dotýkajúcich sa čiernych pixelov vznikne veľké množstvo čiernych obdĺžnikov. Práca vychádza z existujúcich metód slúžiacich na segmentáciu dokumentov. Aby bolo možné segmentovať dashboardy, je potrebné ich vhodným spôsobom predspracovať. V rámci predspracovania sú použité utility ako odstránenie gradientu, posterizácia, prevedenie do stupne sivej a prahovanie. Takto sa získal čierne biely obrázok, na ktorom sa použil algoritmus automatickej segmentácie.

Prínosom práce je vyššia miera automatizácie pri analýze dashboardov. Pokiaľ v iných nástrojoch bolo nutné previesť segmentáciu užívateľom, navrhnutá metóda vykoná segmentáciu automaticky s možnosťou korekcie výsledkov segmentácie. Segmentácia bola založená na výskumu rozpoznávania regiónov užívateľmi. Je využiteľná pre návrhárov dashboardov, ktorý takto rýchlo získajú spätnú väzbu o vhodnosti dashboardu z rôznych pohľadov.

Možným vylepšením je rozpoznávanie vzorov z obrazu, podľa ktorých je možné pospájať časti obrazu s ohľadom na kontext, natrénovanie algoritmu podľa najčastejšie použitých grafických elementov. Prípadne by bolo možné využiť komplikovanejšie prístupy ako napríklad neurónové siete.

Literatúra

- [1] Extensible Markup Language (XML).
URL <https://www.w3.org/XML/>
- [2] Anagnostopoulos, C.-N. E.; Anagnostopoulos, I. E.; Psoroulas, I. D.; aj.: License plate recognition from still images and video sequences: A survey. *IEEE Transactions on intelligent transportation systems*, ročník 9, č. 3, 2008: s. 377–391.
- [3] Dougherty, G.: *Digital image processing for medical applications*. Cambridge University Press, 2009.
- [4] Eckerson, W. W.: *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley & Sons, 2010.
- [5] Few, S.: *Information Dashboard Design*. O'Reilly, 2006, ISBN 0-596-10016-7.
- [6] Ha, J.; Haralick, R. M.; Phillips, I. T.: Recursive XY cut using bounding boxes of connected components. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, ročník 2, IEEE, 1995, s. 952–955.
- [7] Kruatrachue, B.; Moongfangklang, N.; Siriboon, K.: Fast Document Segmentation Using Contour and XY Cut Technique. In *WEC (5)*, 2005, s. 27–29.
- [8] Lebourgeois, F.; Bublinski, Z.; Emptoz, H.: A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, IEEE, 1992, s. 272–276.
- [9] Lee, S.-W.; Ryu, D.-S.: Parameter-free geometric document layout analysis. *IEEE Transactions on pattern analysis and machine intelligence*, ročník 23, č. 11, 2001: s. 1240–1256.
- [10] Nagy, G.; Seth, S.: Hierarchical representation of optically scanned documents. 1984.
- [11] Ngo, D.; Teo, L.; Byrne, J.: Formalising guidelines for the design of screen layouts. *Displays*, ročník 21, č. 1, 2000: s. 3–15.
- [12] Ngo, D. C. L.; Byrne, J. G.: Application of an aesthetic evaluation model to data entry screens. *Computers in Human Behavior*, ročník 17, č. 2, 2001: s. 149–185.
- [13] Ngo, D. C. L.; Teo, L. S.; Byrne, J. G.: Modelling interface aesthetics. *Information Sciences*, ročník 152, 2003: s. 25–46.

- [14] Priyadharshini, N.; Vijaya, M.: Genetic programming for document segmentation and region classification using discipulus. *arXiv preprint arXiv:1303.0460*, 2013.
- [15] Samet, H.: Region representation: Quadtrees from boundary codes. *Communications of the ACM*, ročník 23, č. 3, 1980: s. 163–170.
- [16] Vanderdonckt, J.; Gillo, X.: Visual techniques for traditional and multimedia layouts. In *Proceedings of the workshop on Advanced visual interfaces*, ACM, 1994, s. 95–104.
- [17] Ware, C.: *Information visualization: perception for design*. Elsevier, 2012.
- [18] Zen, M.; Vanderdonckt, J.: Towards an evaluation of graphical user interfaces aesthetics based on metrics. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, IEEE, 2014, s. 1–12.

Prílohy

Zoznam príloh

A	Obsah priloženého pamäťového médiá	54
B	Ukážka XML popisu výsledku segmentácie	55
C	Ukážky automatickej segmentácie	56

Príloha A

Obsah priloženého pamäťového médiá

CD nosič obsahuje:

- `/projekt.pdf` – elektronická verzia tejto práce,
- `/text/` – zdrojové súbory elektronickej verzie tejto práce,
- `/dashapp_segmentation.jar` – spustiteľná binárka aplikácie,
- `/src/` – zdrojové súbory metódy automatickej segmentácie,
- `/data/` – príklady dashboardov pre ukážku metódy,
- `/README` – popis obsahu CD a návod ako použiť aplikáciu.

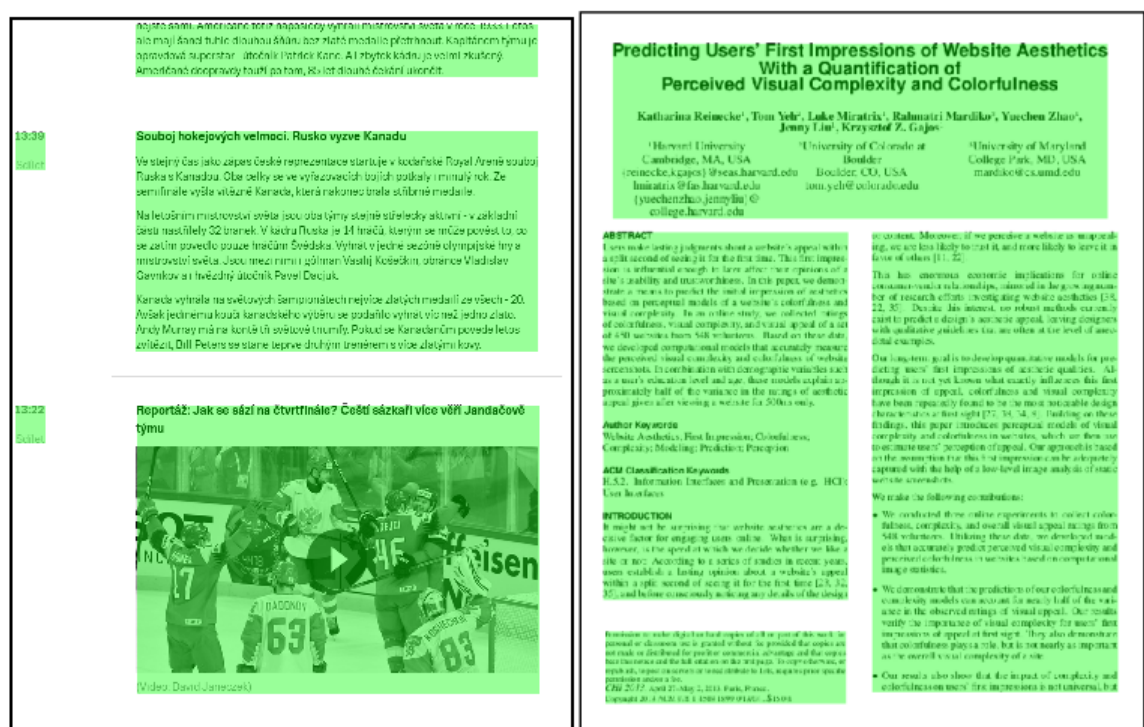
Príloha B

Ukážka XML popisu výsledku segmentácie

```
<dashboard>
  <x>0</x>
  <y>0</y>
  <width>768</width>
  <height>610</height>
  <type>CHART</type>
  <graphicalElement>
    <x>8</x>
    <y>56</y>
    <width>352</width>
    <height>272</height>
    <type>CHART</type>
  </graphicalElement>
  <graphicalElement>
    <x>378</x>
    <y>56</y>
    <width>382</width>
    <height>112</height>
    <type>CHART</type>
  </graphicalElement>
  <graphicalElement>
    <x>380</x>
    <y>474</y>
    <width>380</width>
    <height>121</height>
    <type>CHART</type>
  </graphicalElement>
  <graphicalElement>
    <x>383</x>
    <y>176</y>
    <width>379</width>
    <height>144</height>
    <type>CHART</type>
  </graphicalElement>
  <graphicalElement>
    <x>576</x>
    <y>5</y>
    <width>186</width>
    <height>35</height>
    <type>CHART</type>
  </graphicalElement>
</dashboard>
```

Príloha C

Ukážky automatickej segmentácie



Obr. C.1: Výsledok aplikovania metódy automatickej dekompozície na webovej stránke a na dokumente.



Obr. C.2: Výsledky dekompozície niektorých dashboard.